Knock, Knock. Who's There? On the Security of LG's Knock Codes

Raina Samuel
New Jersey Institute of Technology
res9@njit.edu

Philipp Markert
Ruhr University Bochum
philipp.markert@rub.de

Adam J. Aviv
The George Washington University
aaviv@gwu.edu

Iulian Neamtiu
New Jersey Institute of Technology
ineamtiu@njit.edu

Abstract

Knock Codes are a knowledge-based unlock authentication scheme used on LG smartphones where a user enters a code by tapping or "knocking" a sequence on a 2x2 grid. While a lesser-used authentication method, as compared to PINs or Android patterns, there is likely a large number of Knock Code users; we estimate, 700,000–2,500,000 in the US alone. In this paper, we studied Knock Codes security asking participants in an online study to select codes on mobile devices in three settings: a control treatment, a blocklist treatment, and a treatment with a larger, 2x3 grid. We find that Knock Codes are significantly weaker than other deployed authentication, e.g., PINs or Android patterns. In a simulated attacker setting, 2x3 grids offered no additional security. Blocklisting, on the other hand, was more beneficial, making Knock Codes' security similar to Android patterns. Participants expressed positive perceptions of Knock Codes, yet usability was challenged. SUS values were "marginal" or "ok" across treatments. Based on these findings, we recommend deploying blocklists for selecting a Knock Code because they improve security but have a limited impact on usability perceptions.

1 Introduction

Mobile device unlock authentication has many variations and there have been extensive user-based studies on the security of knowledge-based mobile authentication, including Android graphical unlock patterns [4, 47], PINs [10, 38, 50], as well as using passwords on mobile devices [40]. The conclusion of most of this work is that mobile device users, much like with traditional password selection [18, 28, 39], opt for predictable and easily guessed authenticators. Additionally several physical attacks have been proposed on knowledge-based mobile authentication, such as smudge attacks [6], sensor attacks [7, 12], vision attacks [51], acoustic signals [52], and shoulder surfing [5, 19, 22].

Into this space, LG developed *Knock Codes* as a new mobile authentication system that is designed to combat some of these attacks[1] and provide, per LG's advertising,[2] "perfect security." Knock Codes require a user to recall a pre-selected series of at least 6 and at most 10 knocks[3] (or taps) on a 2×2 quadrant which is displayed upon setup and can be entered with the phone screen on or off. Knock Codes are used less frequently than PINs or Android patterns, but we estimate that there is a large number of Knock Code users, 700,000–2,500,000 in the US alone.

To evaluate the security and usability of Knock Codes, we conducted two online user studies on Amazon Mechanical Turk: a preliminary study ($n = 218$) and a main study ($n = 351$), analyzing a total of 1,138 Knock Codes (436 in the preliminary study and 702 in the main study). In the main study, we evaluated three between-group treatments: a control treatment, where participants used the current 2x2 Knock Code interface; a blocklist treatment, where participants selected 2x2 Knock Codes with some popular codes, as measured in the preliminary study, being disallowed; and finally, a big grid treatment, where participants selected Knock Codes on a larger, 2x3 grid.

We analyzed the selected Knock Codes across treatments and scenarios for security using standard guessing metrics, considering both an offline attacker with unlimited guesses and an online attacker with a limited number of guesses. We find that Knock Codes, as currently deployed, offer *worse* security (51.3 % guessed after 30 attempts) as compared to

USENIX Symposium on Usable Privacy and Security (SOUPS) 2020.
August 9–11, 2020, Virtual Conference.

[1] https://youtu.be/0Imk5JILUc0 (as accessed on June 11, 2020)

[2] https://youtu.be/NRInfu-Lhnc (as accessed on June 11, 2020)

[3] In earlier models, like the 2014 LG G2 [46], where this method first appeared, codes required at least 3 and at most 8. Newer models require 6 to 10 knocks occurring in at least 3 quadrants.

other widely available unlock authentication schemes, e.g., 4-digit PINs (28.0 %), 6-digit PINs (25.4 %) and Android unlock patterns (36.6 %).

While it seems like a straightforward attempt to increase security, an expanded Knock Code grid to 2x3 does not increase, and sometimes worsens, security as compared to 2x2 Knock Codes. After 30 attempts, a simulated attacker correctly guesses *more* 2x3 Knock Codes compared to 2x2 (41 % vs. 37 %). However, blocklisting common Knock Codes (as collected in the preliminary study) is more effective at improving guessing security: only 19 % of these codes were guessed within 30 attempts in simulation.

Overall, participants perceived Knock Codes (across treatments) as secure; however, among all treatments, participants were more hesitant to rate Knock Codes as *more secure* than PINs, Android Unlock Patterns, or alphanumeric passwords. Despite the fact that participants reported Knock Codes as "simple" and "memorable", responses to the SUS [11] questions averaged to "marginal" or "ok" usability (69.8, 68.1, and 64.3, for the control 2x2 treatment, the larger 2x3 treatment, and the blocklist informed 2x2 treatment, respectively). Entry and recall times for Knock Codes were also much slower than what was reported for PINs and Android patterns [27, 38], suggesting lower usability.

Based on the survey and analysis, we make the following contributions and findings:

- We conducted a user study of Knock Codes that considers usability and security analysis.
- We find that Knock Codes, as currently deployed, offer worse security compared to other available methods, both in terms of an online and offline guessing analysis.
- We evaluated different designs for Knock Codes, finding that larger grid sizes offer no benefits (and might actually be less secure) while blocklisting offers promise for improving security.
- We analyzed both qualitative and quantitative feedback of the perceptions of security and usability of Knock Codes, finding that while there are some features of Knock Codes that users like the overall usability was "ok" or "marginal" and the security perceptions were weak compared to other available schemes.

These results indicate that users are interested in new forms of mobile authentication, in particular ones that have options for unlocking with the display off. However, given the usability and security challenges of Knock Codes, *we would not recommend further deployment as currently configured.* For users and developers who wish to continue to use Knock Codes, we would recommend using a blocklist to inform selection as it provides increased security with small effects on usability.

2 Related Work and Background

While Knock Codes have not been broadly studied in the community, other mobile authentication methods have been investigated widely, namely PINs [16, 20], patterns [4, 44, 47], passwords [29, 35], and biometrics [42], as well as adoption rates [27] and authentication times [26].

Research on user-chosen authentication has shown that users tend towards predictable and popular choices, regardless of the authentication method. For instance, Bonneau et al. [10] studied 4-digit PINs and concluded that while 4-digit PINs fare better in user management and choices, guessing the birthday is an effective strategy to access a user's account. Wang et al. showed that 6-digit PINs have marginally better security than 4-digit PINs, yet both English and Chinese users fall into certain patterns when choosing PINs [50].

Markert et al. collected PINs specifically primed for mobile authentication and demonstrated that 6-digit PINs offer little (and perhaps worse) benefit than 4-digit PINs against a throttled attacker. Moreover, non-enforcing blocklists (as deployed by iOS) do not increase security [38]. We use an enforcing blocklist in our data collection, as recommend by Markert et al., and compare Knock Codes to the same RockYou [18] and Amitay [1] datasets used by Wang et al. and Markert et al.

Patterns, or graphical passwords, have been studied in multiple contexts, including smudge attacks [6], shoulder-surfing [5, 19, 23, 37], and user strength perceptions [2, 3]. The selection process has also been studied [4, 44, 47], and in all cases, users choices are predictable. We compare our results to those from Uellenbeck et al. [47] and Aviv et al. [4].

There have also been proposals for incorporating more tactile interaction into mobile authentication. For example, Deyle and Roth suggested using "tactile pins" [21]. Kuber et al. [32–34] studied tactile stimuli: a special mouse with a 4x4 matrix of PINs for selecting a "tactile password." Krombholz et al. considered extra touch interactions through pressure-sensitive touches on iPhones to enhance PINs [31]. However, these user interaction modalities are very different from Knock Codes. Similar to Knock Codes, "personal identifiable chords" (PIC) for smartwatches (a multi-touch PIN entered on a 2x2 grid) have been proposed [41]; these differ in setting (smartwatches) and input type (multi-touch), but the approach could be used to improve Knock Codes by adding multi-touch.

Along with security, usability is an important facet regarding the adoption of authentication methods, thus, quantifying user feedback of such methods is pertinent [43]. Regarding biometric adoption and perceptions, users considered biometrics to be more secure than PINs according to Bhagavatula et al. [8]. In addition, usability factors (such as poor lighting for facial recognition) contributed to users' negative feedback and reluctance to adopt this method versus a more convenient method such as fingerprint recognition. Even with biometrics, this can lead to users choosing weaker forms of knowledge-based authenticators [14].

Figure 1: Screenshot of a video exploring Knock Codes (https://youtu.be/tPYypLe8LEU) where a user enters a Knock Code with the screen off to unlock the phone. This was used to provide instructions and background information to users on Knock Codes.

3 Methodology

We collected data via Amazon Mechanical Turk (MTurk) using an online survey whereby participants were directed to use their mobile devices (checked via the user-agent) to select *two* Knock Codes as well as answer general questions about Knock Codes and their demographics. The two Knock Codes were primed based on different security scenarios, as informed by prior work of Loge et al. [36]. We found some, but minor, differences between Knock Codes in each scenario, similar to Loge et al.'s findings for Android patterns.

We conducted two studies: a preliminary study and a main study which is based on the preliminary study and presented here. The main difference between the two studies is that the main study was focused on participants using mobile devices while the preliminary allowed participants to use traditional computers. From the preliminary study, we were able to refine the main study as well as develop a blocklist of the 30 most common Knock Codes selected in the preliminary study (see Table 3). We provide all study material in the Appendices. Both studies were approved by our institutional review board (IRB).

We found that usage and awareness of Knock Codes are relatively uncommon. Only 3% of our participants in the main study responded that they use Knock Codes, see Table 2 and only 1% reported so in our preliminary study. Despite the low percentages, this suggests that 700K-2.5M users may deploy Knock Codes in the US alone, and we would ideally focus our study just on these users. This is unfortunately not feasible due to the low concentration on MTurk, and as such, we consider a broader set of study participants who may (or may not) be aware of Knock Codes. For those unaware of Knock Codes, our survey would simulate their first experience, as would be the case if they were selecting Knock Codes for the first time on a new device.

Detailed description of the survey. The survey consisted of 12 parts as described below. Please see Appendix A for the exact questions and wording on the pages. We refer to specific questions within a survey part using the page name and question number.

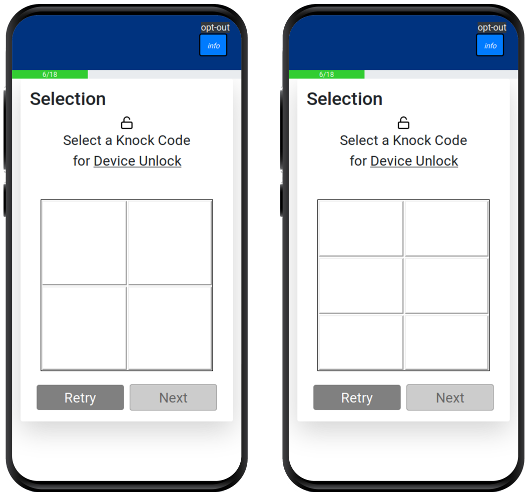
1. *Overview and Informed Consent:* Upon starting the survey, participants were informed about the nature of the research (per the requirements of our IRB), and provided general instructions for proceedings.

2. *Device Usage Questions:* Participants reported on the number of mobile devices (as defined by a smartphone but excluding tablet computers and laptops) they own, the brands they use, and which types of mobile authentication they use on those devices. We use this data, normalized to US census data, to estimate Knock Code usage.

3. *Instructions:* As we could not expect participants to be familiar with Knock Codes, we provided detailed instructions of Knock Codes. This included a GIF animation of a user entering a Knock Code (see Figure 1), a display of the entry screen used later in the survey (see Figure 2), and requirements of Knock Codes (use at least 3 different regions and at least 6 total knocks). We also introduced the size of the grid, 2x2 for participants who were assigned to the control or blocklist treatment, and 2x3 for the group that tested a larger grid. Those in the blocklist treatment were *not* informed of the existence of the blocklist. A detailed description of the treatments is given later in this section.

4. *Practice:* After the instructions, participants could practice selecting a sample Knock Code and familiarize themselves with the interface, before proceeding to the actual Knock Code selection. It was clearly stated that this stage was for practice purposes only. Participants practiced on the appropriate grid size for their treatment and for those in the blocklist treatment, there was no blocklist in place yet, i.e., no indication that a code would or would not be allowed.

5. *Scenario Overview:* In addition to a treatment, each participant was assigned to two scenarios under which they would select Knock Codes for protection. The first of the scenarios was always *Device Unlock*; the other was either *Banking App* or *Shopping Cart*. These scenarios were adapted from prior work of Loge et al. [36] for collecting Android patterns. Participants were made aware of *both* scenarios before proceeding and the order in which they would be asked to select Knock Codes. On this page, we also highlighted that the selected Knock Code will have to be recalled later, hence, participants were asked to "choose something that is secure and memorable."



Selection

Select a Knock Code for Device Unlock

(a) con-2x2 & bl-2x2

Selection

Select a Knock Code for Device Unlock

(b) big-2x3

Figure 2: (a) Interface for selecting 2x2 Knock Codes and (b) interface for selecting 2x3 Knock Codes. Both designs mimic the look and feel of LG's Knock Code implementation.

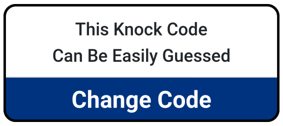

This Knock Code
Can Be Easily Guessed

Figure 3: Blocklist warning display, which mimics blocklist warnings as used by iOS for PINs.

6. *Select and Confirm (2x):* Participants were prompted to select a Knock Code for the scenario, and confirm it before proceeding. The respective pages are shown in Figure 2. Participants of the blocklist treatment saw the warning message shown in Figure 3 if any selection was disallowed. Table 3 contains the list of blocklisted codes as collected in the preliminary study.
7. *Selection Feedback (2x):* After selecting and confirming a Knock Code, participants were asked for feedback about their views on the security of their code and any difficulties in selecting a secure and usable code. Data was collected in both Likert agreement and through open answer forms.
8. *Security Prompts:* Now with more familiarity with Knock Codes, participants answered questions about the perceived security of Knock Codes, and also compared it to PINs and Android Unlock Patterns. Participants also provided qualitative feedback on their security likes and dislikes related to Knock Codes in general.
9. *Usability Prompts:* We asked the 10 System Usability Scale questions [11] related to Knock Codes (plus an attention test).
10. *Recall (2x):* Participants were asked to recall their selected Knock Codes. We allowed up to three guesses for each of the scenarios and forwarded participants if they were not able to recall their Knock Code within this limit.
11. *Demographic Questions:* Participants answered basic demographic questions about their age, gender, dominant hand, educational background, and technology background. We also included another attention check question on this page.
12. *Submission:* The survey ended with participants answering an honesty question (i.e., indicated yes/no to "I honestly participated in this survey and followed instructions completely."). Negative responses were removed from the results, however, all participants were compensated for their work.

Treatments. As part of the study, we assigned participants to one of three treatments. In addition to the standard implementation of LG's Knock Code, which we refer to as **control 2x2** or **con-2x2** throughout this paper, we tested two additional ones.

We first include a blocklist treatment (**blocklist informed 2x2** or **bl-2x2**) which differs from the control 2x2 treatment by the fact that we blocklisted 30 Knock Codes. These codes were the most frequently used as measured in the preliminary study (see Table 3). The blocklist warning, shown in cases of a blocklist hit, is depicted in Figure 3 and is a copy of a warning used by Apple on iOS devices to warn users about an insecure PIN choice.

We conjecture that by disallowing participants from selecting these common codes, the Knock Codes they eventually select would be stronger (harder to guess). There is a risk with blocklists as they may increase frustration during the selection process by having to perform selection multiple times. But as setting up an authentication method is a one-time event, we wished to understand if blocklists can improve the security of Knock Codes.

As another method for increasing security, we considered a modification to the Knock Code interface. The **larger 2x3** treatment (**big-2x3**) uses a 2x3 instead of 2x2 grid and provides participants with more options for creating a Knock Code. Theoretically, this increase makes a substantial difference with 72,520,440 possible 2x3 Knock Codes of length 6-to-10, as compared to 1,384,872 2x2 Knock Codes of similar length. The layout is shown in Figure 2b.

We decided to use a 2x3 grid rather than a horizontal extension (3x2) or making a square (3x3) because of the form factor of the phone's screen, which is taller than it is wide. The 2x3 grid offers a natural extension that fits within the form factor of the screen and mirrors the same interface.

Table 1: Overall demographics of the participants from the main study. Note, zero responses are not shown.

		Male		Female		Other		Total	
		No.	**%**	**No.**	**%**	**No.**	**%**	**No.**	**%**
Age	18 – 24	25	7 %	10	3 %	1	0 %	36	10 %
	25 – 34	131	37 %	64	18 %	2	1 %	197	56 %
	35 – 44	46	13 %	31	9 %	0	0 %	77	22 %
	45 – 54	19	6 %	13	3 %	0	0 %	32	9 %
	55 – 64	2	1 %	6	2 %	0	0 %	8	3 %
	Prefer not to say	0	0 %	0	0 %	1	0 %	1	0 %
Dexterity	Left-handed	31	9 %	14	4 %	0	0 %	45	13 %
	Right-handed	182	52 %	103	29 %	3	1 %	288	82 %
	Ambidextrous	9	3 %	7	2 %	0	0 %	16	5 %
	Prefer not to say	1	0 %	0	0 %	1	0 %	2	0 %
Location	Urban	91	26 %	44	12 %	0	0 %	135	38 %
	Suburban	99	29 %	57	16 %	1	0 %	157	45 %
	Rural	33	9 %	23	7 %	2	0 %	58	17 %
	Prefer not to say	0	0 %	0	0 %	1	0 %	1	0 %
Education	High School	36	10 %	6	2 %	1	0 %	43	12 %
	Some College	45	13 %	25	7 %	0	0 %	70	20 %
	Training	8	3 %	9	3 %	0	0 %	17	6 %
	Associates	22	7 %	17	5 %	1	0 %	40	12 %
	Bachelor's	91	26 %	55	16 %	1	0 %	147	42 %
	Master's	19	6 %	10	2 %	0	0 %	29	8 %
	Professional	1	0 %	1	0 %	0	0 %	2	0 %
	Doctorate	1	0 %	1	0 %	0	0 %	2	0 %
	Prefer not to say	0	0 %	0	0 %	1	0 %	1	0 %
Backgrnd.	Technical	102	30 %	28	8 %	2	0 %	132	38 %
	Non Technical	110	31 %	94	27 %	1	0 %	205	58 %
	Prefer not to say	11	3 %	2	0 %	1	0 %	14	4 %
	Total	223	64 %	124	35 %	4	1 %	351	100 %

Table 2: Answers of the participants from the main study regarding their device usage.

		Male		Female		Other		Total	
		No.	**%**	**No.**	**%**	**No.**	**%**	**No.**	**%**
No. Devices	One device	144	41 %	86	24 %	2	0 %	232	66 %
	Two devices	61	18 %	34	10 %	1	0 %	96	28 %
	Three devices	14	4 %	4	1 %	0	0 %	18	5 %
	Four or more devices	4	1 %	0	0 %	1	0 %	5	1 %
Device Usage	Apple	23	5 %	13	3 %	0	0 %	36	8 %
	Google	26	6 %	11	2 %	0	0 %	37	8 %
	Huawei	9	2 %	4	1 %	1	0 %	13	3 %
	LG	51	11 %	26	6 %	0	0 %	77	22 %
	Motorola	40	9 %	16	4 %	0	0 %	56	13 %
	Samsung	115	25 %	77	17 %	2	0 %	194	43 %
	ZTE	7	1 %	4	1 %	1	0 %	12	2 %
	Miscellaneous	23	5 %	6	1 %	0	0 %	29	6 %
Authentication Usage	4 digit PIN	121	21 %	67	13 %	2	0 %	190	34 %
	6 digit PIN	19	3 %	10	2 %	1	0 %	30	5 %
	6+ digit PIN	12	2 %	5	1 %	0	0 %	17	3 %
	Android pattern	69	12 %	22	4 %	0	0 %	91	16 %
	Knock Code	9	2 %	4	1 %	0	0 %	13	3 %
	Fingerprint	96	17 %	41	7 %	2	0 %	139	24 %
	Facial Recognition	33	6 %	14	3 %	0	0 %	47	9 %
	Other	0	0 %	1	0 %	0	0 %	1	0 %
	No Authentication	17	2 %	20	4 %	1	0 %	38	6 %

Recruitment. The survey was distributed as an Amazon Mechanical Turk task, paying $1.25. On average, it took our participants 8.5 minutes to complete the survey. We ran the survey over the course of two days in June 2019. We recruited 351 participants, each creating two Knock Codes, for a total of 702 selected and confirmed Knock Codes, but also additional Knock Codes that were not confirmed, either due to memorability or the blocklists. We do not consider the practice Knock Codes in our analysis.

The demographics and backgrounds of the participants are listed in Table 1 and 2. As usual for Amazon Mechanical Turk, the participants tended to be younger and predominantly male, but there was diversity in other categories. A number of our participants reported using Knock Codes on their devices as part of their authentication choice. As Knock Codes were a new interface to many participants, our design models the scenario where a user acquires and first uses an LG phone to perform the initial Knock Code set-up.

Estimating US Knock Code Usage. We generalized our participants' device usage and authentication methods based on age and normalized it to the US population using census data [48, 49]. We saw that LG's market share in the US had a range between 8% to 12% among the estimated 285,300,000 smartphone users [17, 45]. Using that, as well as a 95% confidence interval, as our lower and upper bounds, we conclude that there are potentially many Knock Code users: 728,693 to 2,567,207 in the US alone. We believe, though, that the actual adoption rate is most likely on the lower end. While this may be an optimistic estimate, it still suggests that there is a substantial number of Knock Code users in the general public, particularly worldwide.

Even though Knock Codes are not as widely adopted as other traditional methods of mobile authentication, it is still important to study user behavior with real-world, deployed authentication systems. In addition, on Google Play many Knock Code apps can be installed on any Android device, thus not limiting Knock Codes to solely LG devices. For instance, the most highly rated Knock Code app on Android, "Knock Lock," boasts more than 1 million installations and claims that it is an innovative lock screen that "will leave intruders baffled" [30]. This app is just one among the plethora of Knock Code knock-off apps that can be found on Google Play, indicating that this authentication method may have a higher adoption rate and influence on mobile authentication systems than appears initially.

4 Limitations

There are a number of limitations associated with our methodology and survey design. One such limitation is that the survey's recall component occurred within a short time frame with minimal distraction tasks. While we can report on short-term memorability of Knock Codes, we cannot report on the memorability over extended time periods, e.g., days.

However, as a mobile unlock authentication method, users must recall their codes frequently, hence short-term recall is still relevant. The increased use of biometrics, which reduces the number of knowledge-based recalls, confounds the issue though, and more research would be needed to better understand long-term memorability of Knock Codes.

There are also some limitations on how likely the selected Knock Codes would be real Knock Codes of real users. We believe that the simple interface and the nature of the initial device setup suggest that these Knock Codes would be akin to those used on real devices. Most of our participants were unfamiliar with Knock Codes when taking the survey and so would be new users of LG devices setting up their Knock Code for the first time. It should also be noted that a few participants who do use Knock Codes (both in the preliminary study and main study) reported that they reused their Knock Code in the survey.

Nevertheless, we attempted to address this limitation and thus decided to provide different security scenarios for which participants should create Knock Codes. This technique was used by Loge et al. [36] when collecting Android Unlock Patterns. The motivation is that different scenarios, one always being device unlock, will help users to be more careful about their choices, similar to how they may be during device setup. In analyzing the data (Section 6), we did not find significant differences between the Knock Codes selected under each scenario for the bl-2x2 treatment but did see some differences for the con-2x2 and larger 2x3 treatment.

Table 3: Top 30 most frequent Knock Codes from the preliminary study, which were used as the blocklist in the bl-2x2 treatment of the main study.

Rank	Knock Code	No.	%
1	▘ ▝ ▗ ▖ ▘ ▝	28	6.4 %
2	▘ ▝ ▖ ▗ ▘ ▝	25	5.7 %
3	▘ ▗ ▝ ▖ ▘ ▗	19	4.4 %
4	▝ ▘ ▖ ▗ ▝ ▘	7	1.6 %
	▘ ▖ ▗ ▘ ▖ ▗	7	1.6 %
	▘ ▖ ▝ ▗ ▘ ▖	7	1.6 %
	▘ ▘ ▝ ▝ ▗ ▗	7	1.6 %
	▘ ▘ ▝ ▝ ▖ ▖	7	1.6 %
9	▘ ▝ ▖ ▗ ▗ ▖	6	1.4 %
	▘ ▘ ▗ ▗ ▝ ▝	6	1.4 %
11	▘ ▗ ▖ ▝ ▘ ▗	5	1.1 %
	▘ ▗ ▖ ▝ ▘ ▖	5	1.1 %
	▘ ▝ ▖ ▗ ▖ ▗	5	1.1 %
	▘ ▘ ▗ ▗ ▝ ▖	5	1.1 %
	▘ ▘ ▖ ▖ ▗ ▗	5	1.1 %
	▘ ▘ ▝ ▝ ▗ ▗ ▖ ▖	5	1.1 %
	▘ ▘ ▝ ▝ ▗ ▖	5	1.1 %
18	▗ ▘ ▝ ▖ ▖ ▝ ▘ ▗	4	0.9 %
	▝ ▗ ▖ ▘ ▝ ▗	4	0.9 %
	▘ ▝ ▖ ▗ ▖ ▝	4	0.9 %
	▘ ▝ ▖ ▘ ▝ ▖	4	0.9 %
	▘ ▝ ▘ ▝ ▖ ▗	4	0.9 %
23	▖ ▖ ▝ ▝ ▘ ▗	3	0.7 %
	▝ ▝ ▘ ▘ ▗ ▗	3	0.7 %
	▘ ▖ ▝ ▗ ▖ ▗	3	0.7 %
	▘ ▝ ▖ ▗ ▘ ▗	3	0.7 %
	▘ ▝ ▖ ▗ ▘ ▝ ▖ ▗	3	0.7 %
	▘ ▝ ▘ ▝ ▖ ▘	3	0.7 %
	▘ ▘ ▗ ▗ ▖ ▖	3	0.7 %
	▘ ▘ ▘ ▝ ▗ ▖	3	0.7 %

5 Statistics of Knock Codes

The first step in analyzing Knock Codes is to determine the frequency statistics. Table 4 displays the 30 most frequent patterns, combined, across the scenarios for three treatments of the main study. The frequencies which we observed in the preliminary study are shown in Table 3. The preliminary study codes and the con-2x2 codes have a lot of overlap, with 42.0% of the Knock Codes from the preliminary study appearing in the top-30 most frequent codes in the Control 2x2 treatment. This helps justify using the most frequent preliminary study codes as the basis of the blocklist for the bl-2x2 treatment.

Code frequency. The most common Knock Code in our control dataset is ▘ ▝ ▖ ▗ ▘ ▝ ($freq = 6.9\,\%$). It starts in the upper left corner, follows a left-to-right sequence, and is repeated until the minimum length of 6 is reached. We observe a similar strategy for the code 🬀 🬁 🬃 🬇 🬏 🬞 ($freq = 4.6\,\%$) which is the most frequent one in the larger 2x3 treatment. However, participants were able to reach the minimum length without repeating the pattern because of the larger grid.

The second most common Knock Code ▘ ▝ ▗ ▖ ▘ ▝ ($freq = 3.9\,\%$) in the control 2x2 treatment starts in the upper left quadrant, moving clockwise. In contrast to this, 🬀 🬇 🬏 🬞 🬃 🬁 ($freq = 4.2\,\%$), the second most used code in the larger 2x3 treatment, has different attributes: participants proceed diagonally over the grid, going down in a right-left movement for the first diagonal and up in a left-right movement for the second one. The first half of the third most used Knock Code 🬀 🬇 🬏 🬁 🬃 🬞 ($freq = 3.8\,\%$) is identical, yet, it differs at the second diagonal which follows a top-down movement instead of bottom-up.

The third most used Knock Code in the control 2x2 treatment (▘ ▘ ▝ ▝ ▖ ▖, $freq = 3.5\,\%$) pursues a left-to-right sequence again, however, participants used double taps to comply with the required minimum length of 6 knocks.

Participants of the blocklist informed 2x2 treatment used this strategy to an even greater extent: the three most used Knock Codes all contain multiple double taps and 51.0 % of all codes created for this treatment include one or more repeated taps. In contrast to this, only 41.0 % of the codes in the control 2x2 treatment and 29.0 % of the codes in the larger 2x3 treatment contain at least one repeated tap. Moreover, the distribution of Knock Codes in the blocklist informed 2x2 treatment is more equal compared to the other two. The most used Knock Code, ▘ ▘ ▖ ▖ ▝ ▝, occurs in only 2.6 % of the cases and as can be seen in Table 4 the distribution flattens the fastest.

Table 4: Top 30 most frequent Knock Codes in all three treatments.

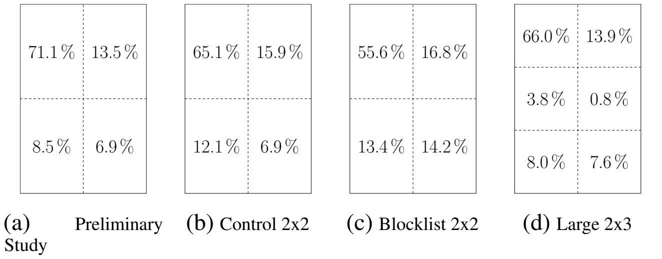
	All Control 2x2				All Blocklist 2x2				All Large 2x3		
Rank	**Knock Code**	**No.**	**%**	**Rank**	**Knock Code**	**No.**	**%**	**Rank**	**Knock Code**	**No.**	**%**
1	▘▝▖▗▘▝	16	6.9 %	1	▘▘▖▖▝▝	6	2.6 %	1	🬀🬁🬃🬇🬏🬞	11	4.6 %
2	▘▝▗▖▘▝	9	3.9 %	2	▝▝▖▖▘▗	5	2.2 %	2	🬀🬇🬏🬞🬃🬁	10	4.2 %
3	▘▘▝▝▖▖	8	3.5 %		▘▘▗▗▝▝▖▖	5	2.2 %	3	🬀🬇🬏🬁🬃🬞	9	3.8 %
4	▘▖▝▗▘▖	6	2.6 %	4	▗▖▘▝▗▖	3	1.3 %		🬀🬃🬏🬞🬇🬁	9	3.8 %
5	▘▗▝▖▘▗	5	2.2 %		▗▝▘▖▗▝	3	1.3 %	5	🬀🬁🬏🬞🬃🬇	8	3.4 %
	▘▘▗▗▝▖	5	2.2 %		▖▘▝▖▘▝	3	1.3 %	6	🬀🬀🬇🬇🬏🬏	7	2.9 %
6	▗▘▖▝▗▘	4	1.7 %		▝▖▘▗▝▖	3	1.3 %		🬀🬃🬏🬁🬇🬞	6	2.5 %
	▖▘▝▝▘▖	4	1.7 %		▘▖▗▗▖▘	3	1.3 %	8	🬁🬇🬞🬏🬃🬀	5	2.1 %
	▘▖▗▝▘▖	4	1.7 %		▘▖▗▝▘▖	3	1.3 %		🬀🬇🬏🬀🬇🬏	5	2.1 %
	▘▝▗▘▝▗	4	1.7 %		▘▖▝▗▘▖▝▗	3	1.3 %		🬀🬁🬞🬏🬃🬇	5	2.1 %
	▘▝▖▗▘▝▖▗	4	1.7 %		▘▝▗▖▗▝	3	1.3 %	11	🬁🬃🬞🬏🬇🬀	4	1.7 %
	▘▘▖▖▗▗	4	1.7 %		▘▝▗▖▘▝▗	3	1.3 %		🬀🬞🬏🬁🬃🬇	4	1.7 %
13	▝▗▖▘▝▗▖▘	3	1.3 %		▘▝▖▗▘▝▖▗▘▝	3	1.3 %		🬀🬀🬃🬃🬏🬏	4	1.7 %
	▝▗▘▖▝▗	3	1.3 %		▘▘▗▗▖▝	3	1.3 %	14	🬞🬃🬁🬀🬇🬏	3	1.3 %
	▝▝▗▗▖▖	3	1.3 %		▘▘▝▝▖▗	3	1.3 %		🬏🬁🬞🬀🬃🬇	3	1.3 %
	▘▗▖▝▘▗▖▝	3	1.3 %	16	▗▗▖▖▝▘	2	0.9 %		🬁🬃🬞🬀🬇🬏	3	1.3 %
	▘▗▖▝▘▗	3	1.3 %		▗▗▖▖▘▘▝▝	2	0.9 %		🬀🬃🬏🬀🬃🬏	3	1.3 %
	▘▗▝▖▘▗▝▖	3	1.3 %		▗▘▖▝▗▘	2	0.9 %		🬀🬃🬁🬇🬏🬞	3	1.3 %
	▘▝▖▗▘▗	3	1.3 %		▖▘▝▗▖▘	2	0.9 %		🬀🬀🬁🬁🬇🬇	3	1.3 %
	▘▝▖▘▝▖	3	1.3 %		▝▖▗▘▝▖	2	0.9 %		🬞🬏🬀🬁🬞🬏	3	1.3 %
	▘▘▗▗▝▝	3	1.3 %		▝▝▖▖▗▗	2	0.9 %	21	🬞🬇🬁🬏🬃🬀	2	0.8 %
	▘▘▖▖▝▝	3	1.3 %		▝▝▘▘▗▗▖▖	2	0.9 %		🬏🬞🬇🬃🬁🬀	2	0.8 %
23	▗▖▗▖▘▝	2	0.9 %		▝▝▘▘▖▖	2	0.9 %		🬏🬇🬀🬁🬃🬞	2	0.8 %
	▖▘▖▘▖▘▝	2	0.9 %		▘▗▝▖▘▗▝▖	2	0.9 %		🬏🬃🬀🬁🬇🬞	2	0.8 %
	▝▖▗▘▝▖	2	0.9 %		▘▖▝▗▗▘	2	0.9 %		🬁🬃🬞🬁🬃🬞	2	0.8 %
	▝▖▘▗▝▖	2	0.9 %		▘▖▝▘▖▝	2	0.9 %		🬁🬁🬇🬇🬞🬞	2	0.8 %
	▝▘▖▗▝▘	2	0.9 %		▘▝▗▖▘▗	2	0.9 %		🬀🬏🬁🬞🬃🬇	2	0.8 %
	▘▗▖▝▗▖	2	0.9 %		▘▝▖▗▝▘	2	0.9 %		🬀🬇🬃🬏🬞🬇	2	0.8 %
	▘▗▖▝▘▘	2	0.9 %		▘▝▖▖▗▘	2	0.9 %		🬀🬃🬏🬞🬞🬞	2	0.8 %
	▘▗▝▖▖▗	2	0.9 %		▘▘▗▖▝▘	2	0.9 %		🬀🬃🬁🬀🬃🬁	2	0.8 %

To summarize, the frequencies of the Knock Codes show different characteristics depending on the assigned treatment, suggesting natural, human tendencies in the selection that can be leveraged in predicting and guessing Knock Codes. We take advantage of this observation when guessing codes. Participants in the blocklist informed 2x2 group use more repeated taps whereas codes created for the 2x3 treatment make use of the larger grid and follow directional patterns. Knock Codes created for the control 2x2 depict a mix and follow both strategies equally.

Start/end quadrant frequency. Figure 4 and 5 present the frequency of start and end taps in the Knock Codes. Clearly, there is a strong tendency to begin codes in the upper-left. Similar observations were made for Android Graphical Patterns [47] and is likely due to the left-to-right nature of the English language which is dominant among our participants. The least common starting points in the preliminary study as well as the control and blocklist treatment were in the lower row. In the larger 2x3 treatment, on the other hand, the middle row is used the least often.

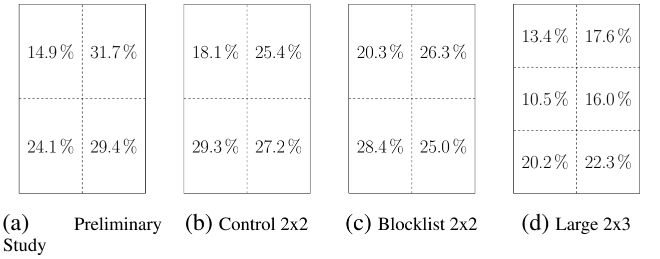
To understand the left/right and up/downshifting of the Knock Codes' start locations we mapped the Cartesian coordinate to each quadrant in the grid, where (-1,1) is the upper left quadrant ▘, (1,1) is the upper right quadrant ▝, (-1,-1) is the lower left quadrant ▖, and (1,-1) is the lower right quadrant ▗. Similarly, in the larger 2x3 treatment, we mapped the coordinates (-1,1), (1,1), (-1,0), (1,0), (-1,-1), and (-1,1) to the grid spaces, scanning left to right, top to bottom. We then computed the average x and y coordinate for the start and end taps, across treatments.

A Shapiro Wilk's test ($p < 0.001$) indicated that the generated frequencies are not normally distributed, so a Mann-Whitney U test was used to identify any initial significance, followed by a posthoc test with Bonferroni correction. We found significant differences between both the control 2x2 and larger 2x3 treatment ($p < 0.001$) as well as blocklist informed 2x2 and larger 2x3 ($p < 0.001$), suggesting that the larger grid size affected how participants chose to start and end their codes.



(a) Preliminary Study (b) Control 2x2 (c) Blocklist 2x2 (d) Large 2x3

Figure 4: Frequency of start quadrants per treatment combined, across all scenarios. More detailed figures with frequencies for every single scenario can be found in the Appendices.



(a) Preliminary Study (b) Control 2x2 (c) Blocklist 2x2 (d) Large 2x3

Figure 5: Frequency of end quadrants per treatment across all scenarios. More detailed figures with frequencies for every single scenario can be found in the Appendices.

Code length. We also analyzed the Knock Codes with respect to length. The average code length was 6.4, 6.5, and 6.2 in each treatment, con-2x2, bl-2x2, and big-2x3, respectively. We observed statistical differences using ANOVA ($f = 11.57, \mathrm{p} < 0.001$) between the treatments. In post hoc analysis, using pairwise *t*-test comparison, the difference lies primarily in the longer big-2x3 Knock Codes, which was statistically different from both bl-2x2 ($p < 0.001$) and the con-2x2 ($p < 0.001$). Surprisingly, the larger grid size encouraged slightly shorter Knock Codes. Regardless, the vast majority of Knock Codes were of length 6, which was the median value, or 8, with a few codes of length 10.

6 Security Analysis

We now analyze the security of Knock Codes. We start by describing the threat model which we are considering for the attack. Afterwards, we analyze the security of Knock Codes by using a perfect knowledge metric in Section 6.1 to define an upper bound on generic attack performance. In Section 6.2, we assess the success rate of a simulated attacker to provide a more realistic security estimation.

Threat Model. We consider a generic, non-targeted attacker that attempts to access an arbitrary victim's device by guessing the Knock Code without additional knowledge or previous observations of the victim. A targeted attacker who may know the victim's tendencies or previously observed an entry (e.g., via a shoulder surfing attack) would likely perform better than the generic attacker. A generic attacker, though, provides a lower bound on the scope of attacker performance, and it also provides a clear comparison point to other reported results [4, 10, 38, 47, 50] which use the same threat model.

For the security analysis, we employ two different attacker variations. First is a *perfect knowledge attacker*, which assumes that the attacker has complete knowledge of the frequency order Knock Codes, from most to least frequent. This attack is still generic as the same strategy is assumed for every victim, and it allows one to estimate the security of the Knock Codes as selected by users. See Section 6.1 for more details.

Second, a *simulated attacker* who knows a subset of the Knock Codes and constructs a model based on that observed distribution. The attacker then attempts to guess a set of arbitrary victims' (unknown) Knock Codes. We use a cross-fold validation to mimic the attacker, whereby the attacker trains on a subset of the data and guesses on an unknown test set.

First-Entry 2x2 Codes. Throughout this section, we refer to a *First-Entry 2x2* dataset which contains participants' first entered codes in the control and blocklist treatment. These codes may or may not have been confirmed (i.e., on the confirm entry screen) either due to lack of recall or because of the blocklist. We include this dataset, as it offers the perspective of an ideal user choice for how the authenticator may have been selected in the absence of external influences. As we expected, this dataset is slighter more secure than that of the confirmed control 2x2 codes and offers insights into how users compromise on security to gain more memorable codes.

6.1 Perfect Knowledge Strength Estimations

We consider the guessing strength of Knock Codes against a perfect knowledge attacker as described by Bonneau et al. [9]. A perfect knowledge attack depicts the upper bound for an attack as it assumes that the attacker knows the attacked dataset and always guesses in the ideal order, that is, the Knock Code with the next highest frequency. This approach has been regularly applied to analyzing mobile authentication, such as Android Patterns [4, 44, 47] or PINs [38, 50].

We use two different perfect-knowledge guessability metrics to evaluate Knock Codes, one based on an offline attack model and one based on an online (or throttled) attack model. An offline attack model assumes that the attacker can guess as many times as possible, while an online attack model assumes an attacker with a limited number of attempts. The online attack model better matches the realities of mobile authentication, where users typically have a maximal number of attempts before the device is locked out. The offline attack model, on the other hand, provides a more holistic approach to measuring the security of a set of user-chosen passwords.

Table 5: Comparison of the guessing metrics for a perfect-knowledge attacker between the treatments and other schemes. A comparison between the scenarios is shown in Appendix B.

Dataset	Online Guessing (Success %)			Offline Guessing (bits)			
	λ_3	λ_{10}	λ_{30}	H_∞	$\widetilde{G}_{0.1}$	$\widetilde{G}_{0.2}$	$\widetilde{G}_{0.5}$
All Control 2x2	14.2 %	28.0 %	51.3 %	3.86	4.20	4.79	5.69
All Blocklist 2x2	6.9 %	16.0 %	35.4 %	5.27	5.79	6.03	6.72
All Large 2x3†	12.9 %	31.5 %	53.4 %	4.40	4.53	4.70	5.54
All First-Entry 2x2†	10.8 %	22.8 %	43.1 %	4.40	4.79	5.35	6.19
3x3 Pattern [4]†	8.6 %	19.4 %	36.6 %	4.69	5.21	5.72	6.76
4x4 Pattern [4]†	7.8 %	18.1 %	32.3 %	5.05	5.47	5.92	7.00
4-digit PINs [1]†	9.5 %	17.2 %	28.0 %	4.40	5.14	6.05	7.21
6-digit PINs [50]†	13.4 %	16.8 %	25.4 %	3.10	3.10	6.38	7.32

†: For a fair comparison we downsampled all marked datasets to the size of Control and Blocklist (232 Knock Codes).

For an offline attack metric, we use *partial guessing entropy* or *α-guesswork* ($\widetilde{G}_\alpha$). Partial guessing entropy estimates the amount of guesswork that is needed to guess a fraction α of all codes. The Min-entropy H_∞ depicts a special case as it is only based on the most frequent Knock Code. As an online (or throttled) attack metric, we use *β-success rate*. It essentially measures what fraction of codes would be guessed if the attacker only had β guesses, e.g., λ_3 considers an attack which is limited to 3 guesses.

Table 5 shows the guessing results for our three treatments as well as the combined dataset First-Entry 2x2. As an additional comparison we included datasets from previous studies for Android patterns [4] as well as 4- and 6- digit PINs [1,50]. Because the datasets all differ in size which would influence the results, we downsampled all marked datasets to the size of control 2x2 and blocklist 2x2 (232 entries) and calculated the statistics for the samples. To rule out any sampling bias, we repeated this process 500 times, removed outliers using Tukey fences with $k = 1.5$, and report the median value of the remaining set in Table 5. With a 95 % confidence level the margin of error is lower than 0.3 % for the online guessing and lower than 0.1 bits for the offline case.

Across all comparisons, we find that Knock Codes in the control 2x2 are significantly weaker in terms of their guessability. This means, Knock Codes as they are currently deployed are more guessable than both 4- and 6-digit PINs as well as Android Patterns. When considering the First-Entry dataset, the differences are less distinct, but even in this ideal case, the inferiority of Knock Codes remains.

Surprisingly, increasing the size of the keyspace by enlarging the grid size to 2x3 offers only little security gain. Moreover, in some cases increasing the grid size may even *decrease* security. This is most apparent when considering a throttled attacker. After 10 guesses, 31.5% of the larger 2x3 codes are guessed compared to 28.0 % for the control 2x2 codes. A similar observation can be made after 30 guesses, 53.4 % of larger 2x3 codes are guessed compared to 51.3 % of control 2x2 codes.

Future works needs to examine why larger Knock Codes performed so poorly, but a similar phenomenon was observed by Aviv et al. with increasing Android patterns from 3x3 to 4x4 grid sizes [4]. Aviv et al. conjectured, and we do so here as well, that there may be a false sense of security that the larger set of choices offers, whereby users believe their individual choice matters less in the face of the increased number of possibilities. Analyzing other grid sizes, such as 3x2 or 3x3, would offer additional insight; nevertheless, it is interesting to see that providing more complexity in how to select Knock Codes does not increase the security.

Finally, we observed strong security improvements with the introduction of a blocklist. As compared to the con-2x2, the blocklist cuts the success rate of an attacker within the first 30 attempts by 30 % to 50 % and increases the guesswork by ~1.5 bits when considering an offline attacker. While the blocklist clearly encouraged more diverse choices, it also had the side effect of increasing user frustration and usability, as we describe later in Section 7.

6.2 Simulated Attacker Strength

We are also interested in modeling a more realistic, limited-knowledge attacker that has access to a subset of training data and attempts to guess some test set of unknown data: a *simulated attacker*.

A simulated attacker must model Knock Codes from a training set to predict a test set. We used a three-gram Markov model probability estimator for the likelihood of a given Knock Code, based on the empirical observations in the test set. This is a standard approach when analyzing user chose secrets, e.g., passwords [13,24], PINs [50], or Android Patterns [4,47]. In order to encode the start and end transitions, we defined special symbols for transitions to ending/starting nodes. This can be defined more formally:

$$x = \{x_{-(g-1)}, \dots, x_{-1}, x_0, x_1, \dots, x_n, \dots, x_{n+g-1}\}$$

where x is the Knock Code of length n with first knock x_1, and g is the gram size. If $i \le 0$ or $i > n$, then this indicates that x_i is a start or end transition state. These extra states are used to capture the early and late transitions taken by a user, for example, for the following Knock Code ▘ ▝ ▖ ▗ , we would produce the following set of tri-grams, where ⊥ is a start state and ⊤ is an end state: (⊥ ⊥ ▘), (⊥▘ ▝), (▘ ▝ ▖), (▝ ▖ ▗), (▖ ▗ ⊤) (▗ ⊤ ⊤).

Using the transition probabilities, as measured in the training data, the attacker can calculate a likelihood measure of a Knock Code by considering the following Markov model formulation,

$$P(x) = P(\mathsf{len}(x)) \cdot P(\mathsf{start}(x)) \cdot P(\mathsf{end}(x)) \cdot \prod_{i=-(g-1)}^{n+(g-1)} P(x_i \dots x_{i+g} \mid x_{i-1} \dots x_{i-1+g}) \qquad (1)$$

Table 6: Guessing performance of a simulated attacker.

		Blocklist Hits		3 Guesses		10 Guesses		30 Guesses	
Dataset	**Codes**	**No.**	**%**	**No.**	**%**	**No.**	**%**	**No.**	**%**
All Control 2x2	**232**	-		**33**	**14 %**	**44**	**19 %**	**85**	**37 %**
Device Unlock	116	-		20	17 %	28	24 %	42	36 %
Banking App.	56	-		0	0 %	4	7 %	8	14 %
Shopping Cart	60	-		9	15 %	11	18 %	23	38 %
All Blocklist 2x2	**232**	**53**	**23 %**	**9**	**4 %**	**14**	**6 %**	**45**	**19 %**
Device Unlock	116	40	35 %	1	1 %	1	1 %	5	4 %
Banking App.	57	8	14 %	3	5 %	3	5 %	3	5 %
Shopping Cart	59	5	9 %	3	5 %	3	5 %	5	9 %
All Large 2x3	**238**	-		**24**	**10 %**	**62**	**26 %**	**97**	**41 %**
Device Unlock	119	-		6	5 %	37	31 %	44	37 %
Banking App.	63	-		1	2 %	6	10 %	15	23 %
Shopping Cart	56	-		5	9 %	10	18 %	15	27 %
All First-Entry 2x2	**464**	-		**42**	**9 %**	**83**	**18 %**	**127**	**27 %**
Device Unlock	232	-		31	13 %	47	20 %	84	36 %
Banking App.	113	-		5	4 %	16	14 %	27	24 %
Shopping Cart	119	-		12	10 %	20	17 %	29	24 %

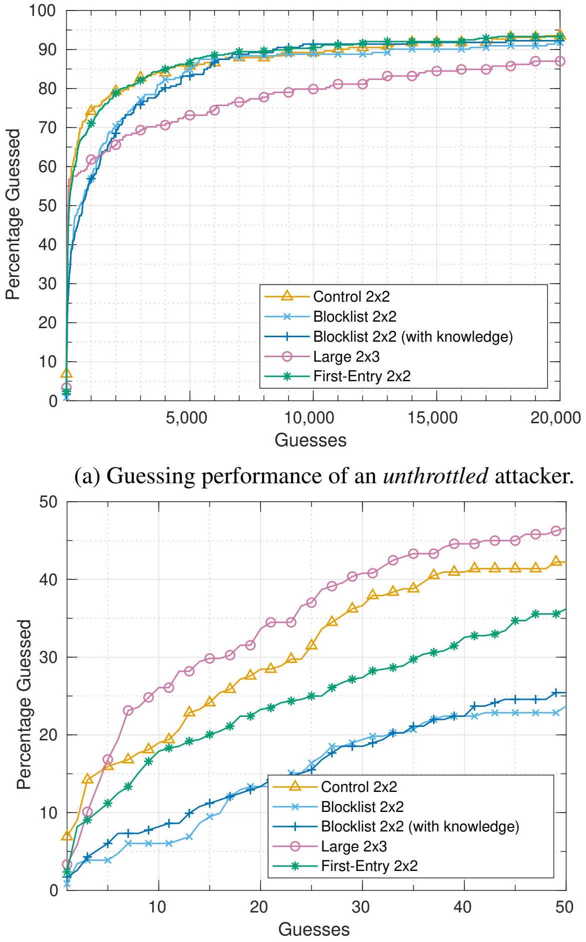
where $P(\cdot)$ is the probability function, $\mathsf{len}(x)$ is the length function, $\mathsf{start}(x)$ is the start function, and $\mathsf{end}(x)$ is the end function. These are our prior probabilities that capture the likelihood of a given length, start quadrant, and end quadrant. The transition probabilities are captured using the conditional probabilities of each transition between each sub-sequence of length g, given the prior state. As not all transitions are represented in our dataset, we used constant smoothing to avoid zero probabilities.

The simulated attackers guessing routine, given a training set, is to (1) create a Markov model of the training data; (2) guess patterns in frequency order of the training set, with ties broken by the likelihood estimation; and (3) guess from a set of additional Knock Codes (not in the training set) sorted based on the likelihood estimation. For (3), we generated a list of all length 6-to-8 Knock Codes for the 2x2 and 2x3 grid sizes, excluding those in our training set that were previously guessed. This accounted for 1,384,872 and 72,520,440 additional 2x2 and 2x3 Knock Codes that could be guessed, respectively. In our blocklist treatment, we assumed the attacker had knowledge of the blocklist.

The results of our simulated attacker are presented in Table 6, and a graphical representation is provided in Figure 6. We report on the average of five randomized cross-fold validations. As expected, the simulated attacker performs worse than the perfect-knowledge attacker, but we find similar results comparing across treatments. Notably, the 2x3 Knock Codes offer little, or worse, security while there is marked improvement for the blocklist informed 2x2 Knock Codes.

7 Usability of Knock Codes

In this section, we focus on the usability metrics of Knock Codes. We first report results on the setup and recall times. Afterwards, we will focus on memorability and recall rates within our study, followed by the qualitative and quantitative responses to security and usability prompts.



(a) Guessing performance of an *unthrottled* attacker.

(b) Guessing performance of a *throttled* attacker.

Figure 6: Guessing performance of a simulated attacker against the different treatments based on the numbers of guesses.

7.1 Setup and Recall Times

Table 7 presents the average selection and recall times, as well as the number of attempts, needed to select a Knock Code. Outliers were removed using Tukey fences with $k = 1.5$.

Participants needed on average 16.2 s and 18.4 s to select and confirm a 2x2 and 2x3 Knock Code, respectively. This is faster than the blocklist treatment (22.5 s), where participants also had to make more attempts due to blocklisting (1.5 vs. 1.1 attempts). In comparison, setting up a 4- or 6-digit PIN takes on average only 7.9 and 11.5 seconds respectively [38] which is significantly faster than Knock Codes. While the described discrepancy between Knock Codes and PINs is distinct, the numbers for PINs may be lower since users are presumably more familiar with PINs as compared to Knock Codes. The differences may decrease with increased familiarity with Knock Codes.

Table 7: The average time and number of attempts required for setup and recall. The standard deviation is shown in brackets.

	Setup			Recall		
Treatment	Time	Attempts	Time/Attempt	Time	Attempts	Time/Attempt
Control 2x2	16.2 *s* (*7.7 s*)	1.1 (*0.4*)	15.6 *s* (*7.6 s*)	8.8 *s* (*4.5 s*)	1.1 (*0.7*)	7.2 *s* (*2.7 s*)
Blocklist 2x2	22.5 *s* (*13.7 s*)	1.5 (*1.1*)	18.3 *s* (*8.6 s*)	11.3 *s* (*6.7 s*)	1.2 (*0.8*)	7.4 *s* (*2.6 s*)
Large 2x3	18.4 *s* (*11.0 s*)	1.1 (*0.5*)	17.4 *s* (*8.4 s*)	8.4 *s* (*4.1 s*)	1.1 (*0.6*)	7.1 *s* (*2.6 s*)

In terms of the recall, which can be compared to unlocking a smartphone, the 2x2 (7.2 s per attempt) and 2x3 Knock Codes (7.1 s per attempt) are more efficient than Knock Codes selected with a blocklist (7.4 s per attempt). With 1.2 attempts per entry, it took participants 11.3 seconds to enter their Knock Codes for the blocklist treatment. Compared to entering an Android pattern (3.0 s) or a PIN (4.7 s) [27], clear usability issues with Knock Codes emerge as entering them is twice as slow. With greater use of Knock Codes, these differences may decrease, but it is unlikely that Knock Codes will be as efficient to enter as patterns or PINs.

7.2 Memorability

We will now go into more details on the memorability as it depicts an important benchmark for any authentication method. We analyzed the memorability of Knock Codes by looking at the recall rates at the end of the survey. While this is an imperfect measure for the memorability, as the survey took most participants less than 10 minutes to complete, it does speak to potential underlying usability issues, particularly if codes were not properly recalled in this short window.

We separated the recall rates based on each treatment. The con-2x2 treatment participants successfully recalled their codes 88.8 % of the time. The participants with the larger 2x3 grid had higher recall rates of 92.9 %, which may suggest an interesting usability vs. security trade-off as this group chose shorter and also some of the weakest Knock Codes. However, we did not find significant differences between the con-2x2 and big-2x3 recall rates using a χ^2 test. We would expect long term memorability rates to be equally high, but further study would be needed to confirm that conjecture.

The worst recall rate came from participants in the bl-2x2 treatment: 80.6 % successfully recalled their Knock Code, and the result was significantly different from the other two recall rates ($p < 0.0001$ for both comparison tests). This could be attributed to the impact of the blocklist, where participants who hit the blocklist had lower recall rates (66.0 %) than those that did not (84.9 %). Most likely, the blocklist affected users in two ways. First, participants who chose blocklisted codes were forced to select multiple codes until landing on one that was not blocklisted. The average number of blocklisting events per user who hit the blocklist was 1.4. Second, that final Knock Code chosen ended up being more complex (as evident in the prior section), and thus harder to recall. Again, this suggests a clear trade-off between usability and security.

We also analyzed the number of attempts to successfully recall a Knock Code. We found no statistical difference across all treatments between the attempts made in recalling the first or second scenario Knock Code correctly. In the big-2x3 treatment and the con-2x2 treatment, participants took on average 1.1 attempts when recalling a Knock Code correctly, with 3 attempts as the maximum. For the bl-2x2 group, users took on average 1.2 attempts to correctly recall a Knock Code, again having a maximum of 3 attempts. Again, we find bl-2x2‘s result to be significantly different in terms of the number of attempts made in the other treatments ($p < 0.001$ vs. big-2x3 and $p < 0.001$ vs. con-2x2), thus showing that the blocklist has an impact on recalling Knock Codes, even for those participants that eventually correctly do so. It is important to note though, that users had a maximum limit of 3 attempts to recall their code before we considered it "cannot be recalled" for the purpose of expediting the survey.

We also analyzed how participants failed to recall their Knock Codes by calculating the average edit distances between the submitted code and the true code for both recalls attempts, one for each scenario. We determined that there was no statistical difference between the average edit distances among treatments. The average edit distance between correct and incorrect recalls was 3.6, suggesting that when users get a code wrong, they get it wrong by a large margin, as the median length Knock Code is 6.

7.3 User Responses

Users provided their opinions and insights regarding Knock Codes' usability and security. We coded these free responses using two independent reviewers where disputes in coding were resolved until consensus was reached. The specific codes and their frequencies are presented in the Appendices.

Overall Knock Codes were perceived positively by users, citing that they were "Easy," "Quick," and "Hard to Guess." The uniqueness of Knock Codes also appealed to users who indicated they especially liked the fact that it is a "Discreet" and "Secure" authentication method which can be inconspicuous and hidden from others.

For many of the participants, this was a new method of authentication, and they employed various tactics when choosing their Knock Codes. We observed such strategies in determining memorable yet secure codes. To make the Knock Code more memorable, the majority of users opted to use some sort of "Pattern" or "Variation" that would be "Sim-

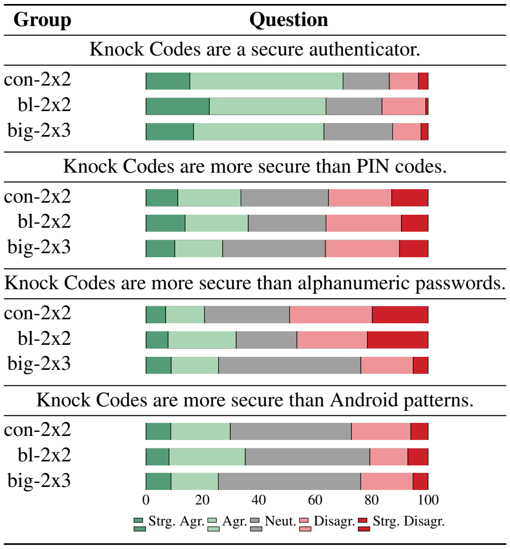


Figure 7: Likert response to comparisons to other mobile authentication methods.

ple." Other techniques users employed include "Directional," "Shape," "Game," and "Repeated." Often, users would create codes based on something "Personal" to them, such as the letter of a word that had meaning to the user.

While many users did not have a specific strategy for security and still focused on making their code "Easy to Remember" as the main priority, others determined that using "All Quadrants" or " Multiple Regions," as well as making the code "Long" or "Random" or being "Unexpected" and "Different" would secure their codes. Making their codes "Hard to Guess" often included attempts to obfuscate the number of clicks and the regions, using speed and potentially unpredictable tactics. Users continued to use similar tactics for memorability to double as security in their Knock Codes, for instance having "Repeated" regions.

Upon comparing Knock Codes with other forms of security, on average users found passwords, PINs, and Android patterns to be more secure than Knock Codes (see Figure 7). Overall, users found Knock Codes adequately secure, i.e., being difficult to hack, resistant to smudge attacks and shoulder surfing. However, they were not completely convinced about Knock Codes' security. Users expressed what they disliked overall, specifically that they found Knock Codes "Hard to Remember" and "Insecure," paving the way for an attacker to easily guess a Knock Code. They also found the interface provided "No improvement" and disliked how it was " Hard to type-in" the Knock Codes.

To have a more general opinion of the overall usability of Knock Codes, we employed the System Usability Scale (SUS). The full Likert responses are found in the Appendices. The average response for the con-2x2 treatment is 69.8, the big-2x3 is 68.1, and the bl-2x2 is 64.3. These scores are generally rated as "ok" or "marginal," with only the control treatment potentially offering some above-average usability.

8 Discussion

As reported above, while most participants offered some positive thoughts, their perception of the security of Knock Codes lagged behind other deployed options, and the SUS values for all schemes were "ok" or "marginal." There was some positive feedback on Knock Codes which suggests an openness to new designs in mobile authentication, particularly to authentication that can be entered while the phone screen is off. There was also increased perceptions of security from targeted attacks, e.g., via shoulder surfing [5, 19, 22]. It is reasonable to view Knock Codes as offering new design concepts that can ultimately improve mobile authentication.

However, we find that Knock Codes, as currently deployed, provide weaker security than other available knowledge-based, mobile unlock methods, such as 4-/6-digit PINs and Android patterns. This is far from the "perfect security" promised by LG's advertisement of Knock Codes. As such, we cannot recommend deploying Knock Codes in their current form as compared to alternative authentication options.

Our results also indicate that a straightforward improvement like increasing the grid size to 2x3 may offer little or worse security. Blocklisting common Knock Codes, on the other hand, does provide more resilience to a throttled attacker, as has been found in password authentication [25] and PINs [38]. Yet, blocklisting runs the risk of increasing user frustration during selection, but since selecting a Knock Code is a one-time event, the usability trade-off of adding a blocklist may be *extremely worthwhile* if Knock Codes continue to be available to LG users. It may also be worthwhile for designers to invest in other methods for improving Knock Code selection, e.g., forcing users to start or end at given quadrants, similar to SysPal [15], or using multi-touch, like chords [41].

9 Conclusion

We performed the first comprehensive user study and security analysis of user-chosen Knock Codes using a three-treatment, between groups study: a control 2x2 treatment, a blocklist 2x2 treatment, and a 2x3 treatment. We find that Knock Codes provide weaker security than other mobile unlock authentication, such as 4-digit PINs, 6-digit PINs, and Android pattern, and that increasing the grid size offered little (or worse) security outcomes, while the addition of a blocklist of common codes substantially increased the security against a throttled attacker. However, Knock Codes suffered in terms of usability, both in terms of entry/recall time and user perception.

Acknowledgments We wish to thank Timothy Forman, Maximilian Golla, and Genry Krichevsky for their assistance. This material is based upon work supported by the National Science Foundation under Grants Nos. 1845300 and 1617584; the research training group "Human Centered Systems Security" (NERD.nrw) sponsored by the state of North Rhine-Westphalia, Germany; and the Army Research Laboratory Cooperative Agreement Number W911NF-13-2-0045 (ARL Cyber Security CRA).

References

[1] Daniel Amitay. Most Common iPhone Passcodes, June 2011. http://danielamitay.com/blog/2011/6/13/most-common-iphone-passcodes, as of June 11, 2020.

[2] Panagiotis Andriotis, Theo Tryfonas, and George Oikonomou. Complexity Metrics and User Strength Perceptions of the Pattern-Lock Graphical Authentication Method. In *Conference on Human Aspects of Information Security, Privacy and Trust*, HAS '14, pages 115–126. Springer, Heraklion, Crete, Greece, June 2014.

[3] Panagiotis Andriotis, Theo Tryfonas, George Oikonomou, and Can Yildiz. A Pilot Study on the Security of Pattern Screen-Lock Methods and Soft Side Channel Attacks. In *ACM Conference on Security and Privacy in Wireless and Mobile Networks*, WiSec '13, pages 1–6, Budapest, Hungary, April 2013. ACM.

[4] Adam J. Aviv, Devon Budzitowski, and Ravi Kuber. Is Bigger Better? Comparing User-Generated Passwords on 3x3 vs. 4x4 Grid Sizes for Android's Pattern Unlock. In *Annual Computer Security Applications Conference*, ACSAC '15, pages 301–310, Los Angeles, California, USA, December 2015. ACM.

[5] Adam J. Aviv, John T. Davin, Flynn Wolf, and Ravi Kuber. Towards Baselines for Shoulder Surfing on Mobile Authentication. In *Annual Conference on Computer Security Applications*, ACSAC '17, pages 486–498, Orlando, Florida, USA, December 2017. ACM.

[6] Adam J. Aviv, Katherine Gibson, Evan Mossop, Matt Blaze, and Jonathan M. Smith. Smudge Attacks on Smartphone Touch Screens. In *USENIX Workshop on Offensive Technologies*, WOOT '10, pages 1–7, Washington, District of Columbia, USA, August 2010. USENIX.

[7] Adam J. Aviv, Benjamin Sapp, Matt Blaze, and Jonathan M. Smith. Practicality of Accelerometer Side Channels on Smartphones. In *Annual Computer Security Applications Conference*, ACSAC '12, pages 41–50, Orlando, Florida, USA, December 2012. ACM.

[8] Chandrasekhar Bhagavatula, Blase Ur, Kevin Iacovino, Su Mon Kywey, Lorrie Faith Cranor, and Marios Savvides. Biometric Authentication on iPhone and Android: Usability, Perceptions, and Influences on Adoption. In *Workshop on Usable Security*, USEC '15, San Diego, California, USA, February 2015. ISOC.

[9] Joseph Bonneau. The Science of Guessing: Analyzing an Anonymized Corpus of 70 Million Passwords. In *IEEE Symposium on Security and Privacy*, SP '12, pages 538–552, San Jose, California, USA, May 2012. IEEE.

[10] Joseph Bonneau, Sören Preibusch, and Ross Anderson. A Birthday Present Every Eleven Wallets? The Security of Customer-Chosen Banking PINs. In *Financial Cryptography and Data Security*, FC '12, pages 25–40, Kralendijk, Bonaire, February 2012. Springer.

[11] John Brooke. SUS: A Quick and Dirty Usability Scale. *Usability Evaluation in Industry*, pages 189–194, 1996.

[12] Liang Cai and Hao Chen. TouchLogger: Inferring Keystrokes on Touch Screen from Smartphone Motion. In *Workshop on Hot Topics in Security*, HotSec '11, Berkeley, California, USA, August 2011. USENIX.

[13] Claude Castelluccia, Markus Dürmuth, and Daniele Perito. Adaptive Password-Strength Meters from Markov Models. In *Symposium on Network and Distributed System Security*, NDSS '12, San Diego, California, USA, February 2012. ISOC.

[14] Ivan Cherapau, Ildar Muslukhov, Nalin Asanka, and Konstantin Beznosov. On the Impact of Touch ID on iPhone Passcodes. In *Symposium on Usable Privacy and Security*, SOUPS '15, pages 257–276, Ottawa, Canada, July 2015. USENIX.

[15] Geumhwan Cho, Jun Ho Huh, Junsung Cho, Seongyeol Oh, Youngbae Song, and Hyoungshick Kim. SysPal: System-Guided Pattern Locks for Android. In *IEEE Symposium on Security and Privacy*, SP '17, pages 338–356, San Jose, California, USA, May 2017. IEEE.

[16] N.L. Clarke and S.M. Furnell. Authentication of Users on Mobile Telephones – A Survey of Attitudes and Practices. *Computers & Security*, 24(7):519–527, October 2005.

[17] Comscore, Inc. Top OEMs - Share of Smartphone Subscribers 3 Month Avg. Ending Nov. 2019 vs. 3 Month Avg. Ending Sep. 2019, September 2019. https://www.comscore.com/Insights/Rankings#tab_mobile_smartphone_oems, as of June 11, 2020.

[18] Nik Cubrilovic. RockYou Hack: From Bad To Worse, December 2009. https://techcrunch.com/2009/12/14/rockyou-hack-security-myspace-facebook-passwords/, as of June 11, 2020.

[19] Alexander De Luca, Marian Harbach, Emanuel von Zezschwitz, Max-Emanuel Maurer, Bernhard Ewald Slawik, Heinrich Hussmann, and Matthew Smith. Now You See Me, Now You Don't: Protecting Smartphone Authentication from Shoulder Surfers. In *ACM Conference on Human Factors in Computing Systems*, CHI '14, pages 2937–2946, Toronto, Ontario, Canada, April 2014. ACM.

[20] Alexander De Luca, Roman Weiss, and Heiko Drewes. Evaluation of Eye-Gaze Interaction Methods for Security Enhanced PIN-Entry. In *Australasian Computer-Human Interaction Conference*, OZCHI '07, pages 199–202, Adelaide, Australia, November 2007. ACM.

[21] Travis Deyle and Volker Roth. Accessible Authentication via Tactile PIN Entry. *Computer Graphics Topics*, 3:24–26, 2006.

[22] Malin Eiband, Mohamed Khamis, Emanuel von Zezschwitz, Heinrich Hussmann, and Florian Alt. Understanding Shoulder Surfing in the Wild:Stories from Users and Observers. In *ACM Conference on Human Factors in Computing Systems*, CHI '17, pages 4254–4265, Denver, Colorado, USA, May 2017. ACM.

[23] Alain Forget, Sonia Chiasson, and Robert Biddle. Shoulder-Surfing Resistance with Eye-Gaze Entry inCued-Recall Graphical Passwords. In *ACM Conference on Human Factors in Computing Systems*, CHI '10, pages 1107–1110, Atlanta, Georgia, USA, April 2010. ACM.

[24] Maximilian Golla and Markus Dürmuth. On the Accuracy of Password Strength Meters. In *ACM Conference on Computer and Communications Security*, CCS '18, pages 1567–1582, Toronto, Ontario, Canada, October 2018. ACM.

[25] Hana Habib, Jessica Colnago, William Melicher, Blase Ur, Sean M. Segreti, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. Password Creation in the Presence of Blocklists. In *Workshop on Usable Security*, USEC '17, San Diego, California, USA, February 2017. ISOC.

[26] Marian Harbach, Alexander De Luca, and Serge Egelman. The Anatomy of Smartphone Unlocking: A Field Study of Android Lock Screens. In *ACM Conference on Human Factors in Computing Systems*, CHI '16, pages 4806–4817, San Jose, California, USA, May 2016. ACM.

[27] Marian Harbach, Emanuel von Zezschwitz, Andreas Fichtner, Alexander De Luca, and Matthew Smith. It's a Hard Lock Life: A Field Study of Smartphone (Un)Locking Behavior and Risk Perception. In *Symposium on Usable Privacy and Security*, SOUPS '14, pages 213–230, Menlo Park, California, USA, July 2014. USENIX.

[28] Patrick Kelley, Saranga Kom, Michelle L. Mazurek, Rich Shay, Tim Vidas, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, and Julio López. Guess Again (and Again and Again): Measuring Password Strength by Simulating Password-Cracking Algorithms. In *IEEE Symposium on Security and Privacy*, SP '12, pages 523–537, San Jose, California, USA, May 2012. IEEE.

[29] Daniel V. Klein. "Foiling the Cracker": A Survey Of, and Improvements To, Password Security. In *UNIX Security Workshop*, UNIX '90, pages 5–14, Portland, Oregon, USA, August 1990. USENIX.

[30] Knock Lock. Knock Lock Screen - Applock, 2020. https://play.google.com/store/apps/details?id=com.knocklock.applock&hl=en_US, as of June 11, 2020.

[31] Katharina Krombholz, Thomas Hupperich, and Thorsten Holz. Use the Force: Evaluating Force-Sensitive Authentication for Mobile Devices. In *Symposium on Usable Privacy and Security*, SOUPS '16, pages 207–219, Denver, Colorado, USA, July 2016. USENIX.

[32] Ravi Kuber and Shiva Sharma. Toward Tactile Authentication for Blind Users. In *ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '10, pages 289–290, Orlando, Florida, USA, October 2010. ACM.

[33] Ravi Kuber and Wai Yu. Feasibility Study of Tactile-Based Authentication. *International Journal of Human-Computer Studies*, 68(3):158–181, March 2010.

[34] Ravi Kuber and Wai Yu. Toward Tactile Authentication for Blind Users. In *International Conference on Human Haptic Sensing and Touch Enabled Computer Applications*, EuroHaptics '10', pages 314–319, Amsterdam, Netherlands, July 2010. Springer.

[35] William C. Lindsey and Chak Ming Chie. A Survey of Digital Phase-Locked Loops. *Proceedings of the IEEE*, 69(4):410–431, April 1981.

[36] Marte Løge, Markus Dürmuth, and Lillian Røstad. On User Choice for Android Unlock Patterns. In *European Workshop on Usable Security*, EuroUSEC '16, Darmstadt, Germany, July 2016. ISOC.

[37] Shushuang Man, Dawei Hong, and Manton Matthews. A Shoulder-Surfing Resistant Graphical Password Scheme – WIW. In *International Conference on Security and Management*, SAM '03, pages 105–111, Las Vegas, Nevada, USA, June 2003. CSREA Press.

[38] Philipp Markert, Daniel V. Bailey, Maximilian Golla, Markus Dürmuth, and Adam J. Aviv. This PIN Can Be Easily Guessed: Analyzing the Security of Smartphone Unlock PINs. In *IEEE Symposium on Security and Privacy*, SP '20, pages 1525–1542, San Francisco, California, USA, May 2020. IEEE.

[39] Michelle L. Mazurek, Saranga Komanduri, Timothy Vidas, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, Patrick Gage Kelley, Richard Shay, and Blase Ur. Measuring Password Guessability for an Entire University. In *Conference on Computer and Communications Security*, CCS '13, pages 173–186, Berlin, Germany, October 2013. ACM.

[40] William Melicher, Darya Kurilova, Sean M. Segreti, Pranshu Kalvani, Richard Shay, Blase Ur, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, and Michelle L. Mazurek. Usability and Security of Text Passwords on Mobile Devices. In *ACM Conference on Human Factors in Computing Systems*, CHI '16, pages 527–539, Santa Clara, California, USA, May 2016. ACM.

[41] Ian Oakley, Jun Ho Huh, Junsung Cho, Geumhwan Cho, Rasel Islam, and Hyoungshick Kim. The Personal Identification Chord: A Four Button Authentication System for Smartwatches. In *ACM Asia Conference on Computer and Communications Security*, ASIA CCS '18, pages 75–87, Incheon, Republic of Kore, June 2018. ACM.

[42] Salil Prabhakar, Sharath Pankanti, and Anil K Jain. Biometric Recognition: Security and Privacy Concerns. *IEEE Security & Privacy*, 1(2):63–69, March 2003.

[43] Florian Schaub, Ruben Deyhle, and Michael Weber. Password Entry Usability and Shoulder Surfing Susceptibility on Different Smartphone Platforms. In *International Conference on Mobile and Ubiquitous Multimedia*, MUM '12, pages 13:1–13:10, Ulm, Germany, December 2012. ACM.

[44] Youngbae Song, Geumhwan Cho, Seongyeol Oh, Hyoungshick Kim, and Jun Ho Huh. On the Effectiveness of Pattern Lock Strength Meters: Measuring the Strength of Real World Pattern Locks. In *ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 2343–2352, Seoul, Republic of Korea, April 2015. ACM.

[45] Team Counterpoint. US Smartphone Market Share: By Quarter, November 2019. https://www.counterpointresearch.com/us-market-smartphone-share/, as of June 11, 2020.

[46] Kevin C. Tofel. LG G2 and G Flex Phones Getting the Knock Code Wake and Unlock Feature, March 2014. https://gigaom.com/2014/03/25/lg-g2-and-g-flex-phones-getting-the-knock-code-wake-and-unlock-feature/, as of June 11, 2020.

[47] Sebastian Uellenbeck, Markus Dürmuth, Christopher Wolf, and Thorsten Holz. Quantifying the Security of Graphical Passwords: The Case of Android Unlock Patterns. In *ACM Conference on Computer and Communications Security*, CCS '13, pages 161–172, Berlin, Germany, November 2013. ACM.

[48] U.S. Census Bureau, Population Division. Annual Estimates of the Resident Population by Single Year of Age and Sex for the United States: April 1, 2010 to July 1, 2018 , 2018 Population Estimates, June 2019. https://factfinder.census.gov/bkmk/table/1.0/en/PEP/2018/PEPSYASEXN?#, as of June 11, 2020.

[49] U.S. Census Bureau, Population Division. Annual Estimates of the Resident Population for Selected Age Groups by Sex for the United States, States, Counties, and Puerto Rico Commonwealth and Municipios: April 1, 2010 to July 1, 2018, June 2019. https://factfinder.census.gov/bkmk/table/1.0/en/PEP/2018/PEPAGESEX?#, as of June 11, 2020.

[50] Ding Wang, Qianchen Gu, Xinyi Huang, and Ping Wang. Understanding Human-Chosen PINs: Characteristics, Distribution and Security. In *ACM Asia Conference on Computer and Communications Security*, ASIA CCS '17, pages 372–385, Abu Dhabi, United Arab Emirates, April 2017. ACM.

[51] Guixin Ye, Zhanyong Tang, Dingyi Fang, Xiaojiang Chen, Kwang In Kim, Ben Taylor, and Zheng Wang. Cracking Android Pattern Lock in Five Attempts. In *Symposium on Network and Distributed System Security*, NDSS '17, San Diego, California, USA, February 2017. ISOC.

[52] Man Zhou, Qian Wang, Jingxiao Yang, Qi Li, Feng Xiao, Zhibo Wang, and Xiaofen Chen. PatternListener: Cracking Android Pattern Lock Using Acoustic Signals. In *ACM Conference on Computer and Communications Security*, CCS '18, pages 1775–1787, Toronto, Ontario, Canada, October 2018. ACM.

APPENDICES

A Survey Material

A.1 Main Study

1. Device Usage Questions
When referring to "mobile devices" throughout this survey, consider these to include smartphones and tablet computers, such as iPhone and Android phones and tablets. Traditional laptop computers, two-in-one computers, like the Microsoft Surface, or e-readers, like the Amazon Kindle, are not considered mobile devices for the purposes of this survey.

1. How many mobile devices do you use regularly? (Including phones and tablets, but excluding laptops)
   ○ 0 ○ 1 ○ 2 ○ 3 ○ 4+ ○ Prefer not to say

2. What brand of smartphone do you use? (Select all that apply)
   □ Apple □ Samsung □ LG □ Google (Pixel/Nexus)
   □ Motorola □ ZTE □ I do not own a smartphone
   □ Other: ____________

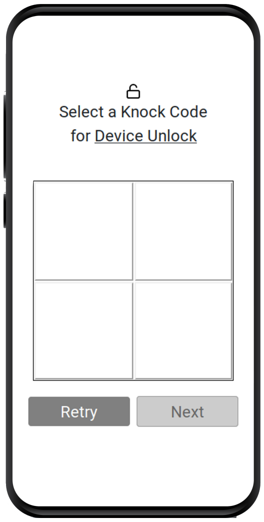
3. Select "No" as the answer to this questions:
   ○ Yes ○ No ○ Sometimes ○ Always

4. Which method(s) do you use to lock your mobile device(s)?(Select all that apply)
   □ 4-digit PIN □ 6-digit PIN □ PIN of other length
   □ Android Graphical Pattern □ LG Knock Codes
   □ Fingerprint □ Face □ None □ Other: ____________

Where indicated, the text and the graphics on the following pages changed depending on the assigned grid size.

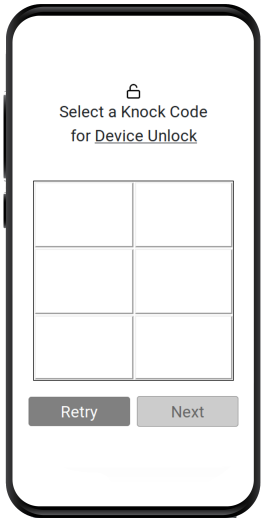
2. What are Knock Codes?
Knock Codes are an authentication method used to unlock your smartphone, much like a PIN. To unlock the phone, the user enters their knock Code by tapping different regions (or quadrants) of a [2x2|2x3] grid on the smartphone display. The grid may or may not be displayed at the time of entry, for example, below is a video of someone entering a Knock Code without a grid displayed.

As part of this survey, you will be asked to select your own Knock Codes using an on-screen approximation of a smartphone. You will enter your codes by clicking on different regions of a [2x2|2x3] grid with your mouse. Below is an image of the [2x2|2x3] grid and smartphone approximation.



Select a Knock Code for Device Unlock

con-2x2 & bl-2x2



Select a Knock Code for Device Unlock

big-2x3

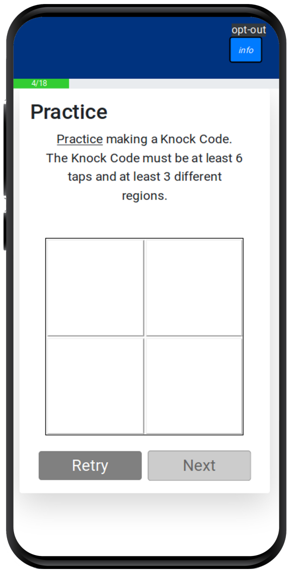
There are some rules! When selecting a Knock Code it must:

1. Use *at least* 3 regions of the grid.
2. Use *at least* 6 total knocks.

On the next page, you will have a chance to practice entering Knock Codes after which you will proceed with the rest of the survey.

Participants performed a practice run of using the interface. After completion, they were given the option to continue.

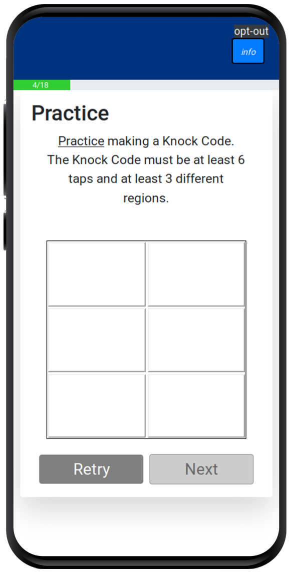
3. Practice



Practice

Practice making a Knock Code. The Knock Code must be at least 6 taps and at least 3 different regions.

con-2x2 & bl-2x2



Practice

Practice making a Knock Code. The Knock Code must be at least 6 taps and at least 3 different regions.

big-2x3

4. Scenarios

For the remainder of this survey, you will be asked to create Knock Codes for different scenarios.

Importantly, you will need to recall these codes later. So choose something that is secure and memorable. However, we ask that you DO NOT write down your codes or use other aids to help you remember.

I understand that I should not write down my codes or use other aids to assist in the survey:

○ I understand

You will be asked to create Knock-Knock Codes for the following scenarios.

All participants were assigned to Device Unlock, and then one of either Banking App or Shopping Cart.

🔓 - *Device Unlock* Create a Knock Code you would use to **unlock your smartphone or tablet**.

💼 - *Banking App* Create a Knock Code you would use to secure access to your **mobile banking application**.

🛒 - *Shopping Cart* Create a Knock Code you would use to protect your **Amazon shopping cart**.

I understand that I should not write down my codes or use other aids to assist in the survey:

○ I understand

Steps 5, 6, and 7 were done twice. First for the Device Unlock, then for the banking or shopping scenario.

5. Selection

Select a Knock Code for [SCENARIO]

6. Confirmation

Confirm the Knock Code for [SCENARIO]

7. Thinking about the Knock Code you just chose, answer the following questions.

1. I feel this Knock Code provides adequate security for this scenario.
   ○ Strongly Agree ○ Agree ○ Neither agree nor disagree
   ○ Disagree ○ Strongly disagree

2. It was difficult to choose this Knock Code.
   ○ Strongly Agree ○ Agree ○ Neither agree nor disagree
   ○ Disagree ○ Strongly disagree

3. What strategy did you use to make your code **more secure**?
   Answer: ____________________

4. What strategy did you use to make your code **more memorable**?
   Answer: ____________________

8. Please answer the following questions/prompts.

Select your agreement/disagreement with the following statements

1. Knock Codes are a secure authenticator.
   ○ Strongly Agree ○ Agree ○ Neither agree nor disagree
   ○ Disagree ○ Strongly disagree

2. Knock Codes are more secure than PIN codes.
   ○ Strongly Agree ○ Agree ○ Neither agree nor disagree
   ○ Disagree ○ Strongly disagree
   ○ Do not know what a PIN code is

3. Knock Codes are more secure than alphanumeric passwords.
   ○ Strongly Agree ○ Agree ○ Neither agree nor disagree
   ○ Disagree ○ Strongly disagree
   ○ Do not know what an alphanumeric passwords is

4. Knock Codes are more secure than Android patterns.
   ○ Strongly Agree ○ Agree ○ Neither agree nor disagree
   ○ Disagree ○ Strongly disagree
   ○ Do not know what an Android pattern is

   Provide general feedback on the following questions

5. What are some aspects you **like** about Knock Codes? (use N/A if you do not wish to answer)
   Answer: ____________________

6. What are some aspects you **do not like** about Knock Codes? (use N/A if you do not wish to answer)
   Answer: ____________________

9. Please answer the following questions/prompts.

Select your agreement/disagreement with the following statements

1. I think that I would like to use Knock Codes frequently.
   ○ Strongly Agree ○ Agree ○ Neither agree nor disagree
   ○ Disagree ○ Strongly disagree

2. I found Knock Codes unnecessarily complex.
   ○ Strongly Agree ○ Agree ○ Neither agree nor disagree
   ○ Disagree ○ Strongly disagree

3. I thought Knock Codes were easy to use.
   ○ Strongly Agree ○ Agree ○ Neither agree nor disagree
   ○ Disagree ○ Strongly disagree

4. I think that I would need the support of a technical person to be able to use Knock Codes.
   ○ Strongly Agree ○ Agree ○ Neither agree nor disagree
   ○ Disagree ○ Strongly disagree

5. I found the various functions in Knock Codes were well integrated.
   ○ Strongly Agree ○ Agree ○ Neither agree nor disagree
   ○ Disagree ○ Strongly disagree

6. I thought there was too much inconsistency in Knock Codes.
○ Strongly Agree ○ Agree ○ Neither agree nor disagree
○ Disagree ○ Strongly disagree

7. I would imagine that most people would learn to use Knock Codes very quickly.
○ Strongly Agree ○ Agree ○ Neither agree nor disagree
○ Disagree ○ Strongly disagree

8. Select Agree as the answer to this question.
○ Strongly Agree ○ Agree ○ Neither agree nor disagree
○ Disagree ○ Strongly disagree

9. I found Knock Codes very cumbersome to use.
○ Strongly Agree ○ Agree ○ Neither agree nor disagree
○ Disagree ○ Strongly disagree

10. I felt very confident using Knock Codes.
○ Strongly Agree ○ Agree ○ Neither agree nor disagree
○ Disagree ○ Strongly disagree

11. I needed to learn a lot of things before I could get going with Knock Codes.
○ Strongly Agree ○ Agree ○ Neither agree nor disagree
○ Disagree ○ Strongly disagree

10. Recall
Recall your Knock Code for [SCENARIO]

11. Demographic Questions

1. What is your age range:
○ 18-24 ○ 25-29 ○ 30-34 ○ 35-39 ○ 40-44 ○ 45-49
○ 50-54 ○ 55-59 ○ 60-64 ○ 65+ ○ Prefer not to say

2. With what gender do you identify:
○ Male ○ Female ○ Non-Binary/Third Gender
○ Not described here ○ Prefer not to say

3. What is your dominant hand?
○ Left handed ○ Right handed ○ Ambidextrous
○ Prefer not to say

4. Where you live is best described as:
○ Urban ○ Suburban ○ Rural ○ Prefer not to say

5. What is the highest degree or level of school you have completed?
○ Some high school ○ High school ○ Some college
○ Trade, technical, or vocational training ○ Associate's Degree ○ Bachelor's Degree ○ Master's Degree
○ Professional Degree ○ Doctorate ○ Prefer not to say

6. Which of the following best describes your educational background or job field?
○ I have an education in, or work in, the field of computer science, computer engineering or IT.
○ I do not have an education in, nor do I work in, the field of computer science, computer engineering, or IT.
○ Prefer not to say

Please indicate if you've honestly participated in this survey and followed instructions completely. You will not be penalized/rejected for indicating "No" but your data may not be included in the analysis:
○ Yes ○ No

A.2 Preliminary Study

1. Demographic Questions

1. What is your age range:
○ 18-24 ○ 25-29 ○ 30-34 ○ 35-39 ○ 40-44 ○ 45-49
○ 50-54 ○ 55-59 ○ 60-64 ○ 65+ ○ Prefer not to say

2. With what gender do you identify:
○ Male ○ Female ○ Non-Binary/Third Gender
○ Not described here ○ Prefer not to say

3. What is your dominant hand?
○ Left handed ○ Right handed ○ Ambidextrous
○ Prefer not to say

4. Where you live is best described as:
○ Urban ○ Suburban ○ Rural ○ Prefer not to say

5. What is the highest degree or level of school you have completed?
○ Some high school ○ High school ○ Some college
○ Trade, technical, or vocational training ○ Associate's Degree ○ Bachelor's Degree ○ Master's Degree
○ Professional Degree ○ Doctorate ○ Prefer not to say

6. Which of the following best describes your educational background or job field?
○ I have an education in, or work in, the field of computer science, computer engineering or IT.
○ I do not have an education in, nor do I work in, the field of computer science, computer engineering, or IT.
○ Prefer not to say

2. Device Usage Questions
When referring to "mobile devices" throughout this survey, consider these to include smartphones and tablet computers, such as iPhone and Android phones and tablets. Traditional laptop computers, two-in-one computers, like the Microsoft Surface, or e-readers, like the Amazon Kindle, are not considered mobile devices for the purposes of this survey.

1. How many mobile devices do you use regularly? (Including phones and tablets, but excluding laptops)
○ 0 ○ 1 ○ 2 ○ 3 ○ 4+

2. What brand of smartphone do you use? (Select all that apply)
□ Apple □ Samsung □ LG □ Google (Pixel/Nexus)
□ Motorola □ ZTE □ Other: ____________

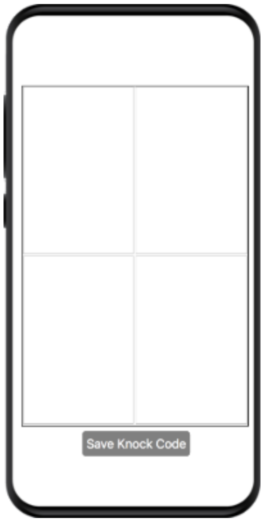
3. Select "No" as the answer to this questions:
○ Yes ○ No ○ Sometimes ○ Always

4. Which method(s) do you use to lock your mobile device(s)?(Select all that apply)
□ 4-digit PIN □ 6-digit PIN □ PIN of other length
□ Android Graphical Pattern □ LG Knock Codes
□ Fingerprint □ Face □ Other: ____________

3. What are Knock Codes?

Knock Codes are an authentication method used to unlock your smartphone, much like a PIN. To unlock the phone, the user enters their knock Code by tapping different regions (or quadrants) of a 2x2 grid on the smartphone display. The grid may or may not be displayed at the time of entry, for example, below is a video of someone entering a Knock Code without a grid displayed.

As part of this survey, you will be asked to select your own Knock Codes using an on-screen approximation of a smartphone. You will enter your codes by clicking on different regions of a 2x2 grid with your mouse. Below is an image of the 2x2 grid and smartphone approximation.



There are some rules! When selecting a Knock Code it must:

1. Use *at least* 3 regions of the grid.
2. Use *at least* 6 total knocks.

On the next page, you will have a chance to practice entering Knock Codes after which you will proceed with the rest of the survey.

4. Practice

Participants performed a practice run of using the interface. After completion, they were given the option to continue.

5. Scenarios

For the remainder of this survey, you will be asked to create Knock Codes for different scenarios.

Importantly, you will need to recall these codes later. So choose something that is secure and memorable. However, we ask that you DO NOT write down your codes or use other aids to help you remember.

I understand that I should not write down my codes or use other aids to assist in the survey:
○ I understand

You will be asked to create Knock-Knock Codes for the following scenarios.

All participants created Device Unlock, and then one of either Banking App or Shopping Cart. The order was randomized.

🔓 - *Device Unlock* Create a Knock Code you would use to **unlock your smartphone or tablet**.
💼 - *Banking App* Create a Knock Code you would use to secure access to your **mobile banking application**.
🛒 - *Shopping Cart* Create a Knock Code you would use to protect your **Amazon shopping cart**.

I understand that I should not write down my codes or use other aids to assist in the survey:
○ I understand

Steps 5, 6, and 7 were done twice. For the Device Unlock and for the banking or shopping scenario. The order was randomized.

6. Selection

Select a Knock Code for [SCENARIO]

7. Confirmation

Confirm the Knock Code for [SCENARIO]

8. Thinking about the Knock Code you just chose, answer the following questions.

1. I feel this Knock Code provides adequate security for this scenario.
   ○ Strongly Agree ○ Agree ○ Neither agree nor disagree
   ○ Disagree ○ Strongly disagree

2. It was difficult to choose this Knock Code.
   ○ Strongly Agree ○ Agree ○ Neither agree nor disagree
   ○ Disagree ○ Strongly disagree

3. What strategy did you use to make your code **more secure**?
   Answer: __________________

4. What strategy did you use to make your code **more memorable**?
   Answer: __________________

9. Please answer the following questions/prompts.
Select your agreement/disagreement with the following statements

1. Knock Codes are a secure authenticator.
   ○ Strongly Agree ○ Agree ○ Neither agree nor disagree
   ○ Disagree ○ Strongly disagree

2. Knock Codes are more secure than PIN codes.
   ○ Strongly Agree ○ Agree ○ Neither agree nor disagree
   ○ Disagree ○ Strongly disagree
   ○ Do not know what a PIN code is

3. Knock Codes are more secure than alphanumeric passwords.
   ○ Strongly Agree ○ Agree ○ Neither agree nor disagree
   ○ Disagree ○ Strongly disagree
   ○ Do not know what an alphanumeric passwords is

4. Knock Codes are more secure than Android patterns.
   ○ Strongly Agree ○ Agree ○ Neither agree nor disagree
   ○ Disagree ○ Strongly disagree
   ○ Do not know what an Android pattern is

   Provide general feedback on the following questions

5. What are some aspects you **like** about Knock Codes? (use N/A if you do not wish to answer)
   Answer: __________________

6. What are some aspects you **do not like** about Knock Codes? (use N/A if you do not wish to answer)
   Answer: __________________

10. Please answer the following questions/prompts.
Select your agreement/disagreement with the following statements

1. I would like to use Knock Codes frequently.
   ○ Strongly Agree ○ Agree ○ Neither agree nor disagree
   ○ Disagree ○ Strongly disagree

2. Knock Codes are unnecessarily complex.
   ○ Strongly Agree ○ Agree ○ Neither agree nor disagree
   ○ Disagree ○ Strongly disagree

3. Knock Codes are easy to use.
   ○ Strongly Agree ○ Agree ○ Neither agree nor disagree
   ○ Disagree ○ Strongly disagree

4. I would need the support of a technical person to be able to use Knock Codes.
   ○ Strongly Agree ○ Agree ○ Neither agree nor disagree
   ○ Disagree ○ Strongly disagree

5. I would make a lot of mistakes if I were to use Knock Codes.
   ○ Strongly Agree ○ Agree ○ Neither agree nor disagree
   ○ Disagree ○ Strongly disagree

6. Most people would learn to use Knock Codes very quickly.
   ○ Strongly Agree ○ Agree ○ Neither agree nor disagree
   ○ Disagree ○ Strongly disagree

7. Select Agree as the answer to this question.
   ○ Strongly Agree ○ Agree ○ Neither agree nor disagree
   ○ Disagree ○ Strongly disagree

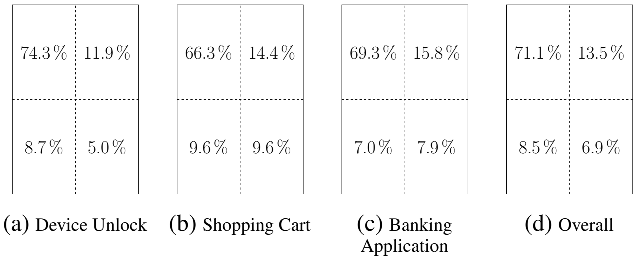
8. I found Knock Codes very cumbersome to use.
   ○ Strongly Agree ○ Agree ○ Neither agree nor disagree
   ○ Disagree ○ Strongly disagree

9. I would neet to practice Knock Codes more before I could use them.
   ○ Strongly Agree ○ Agree ○ Neither agree nor disagree
   ○ Disagree ○ Strongly disagree

11. Recall
Recall your Knock Code for [SCENARIO]

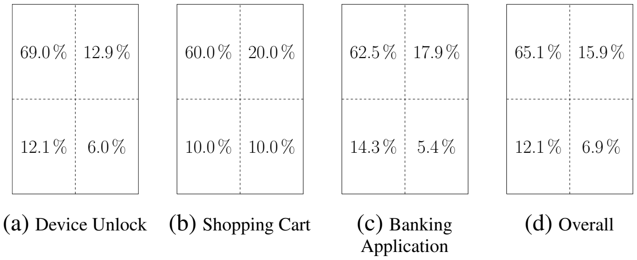
Please indicate if you've honestly participated in this survey and followed instructions completely. You will not be penalized/rejected for indicating "No" but your data may not be included in the analysis:
○ Yes ○ No

B Additional Figures & Tables



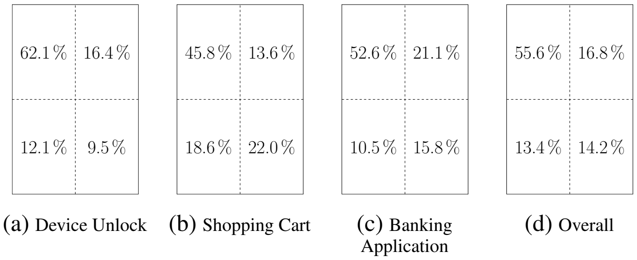
(a) Device Unlock (b) Shopping Cart (c) Banking Application (d) Overall

Figure 8: Frequency of start quadrants per scenario in the preliminary study.



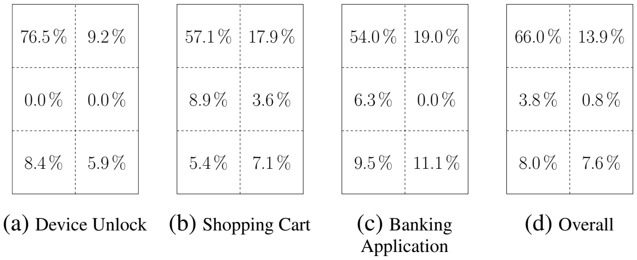
(a) Device Unlock (b) Shopping Cart (c) Banking Application (d) Overall

Figure 10: Frequency of start quadrants per scenario for the control treatment.



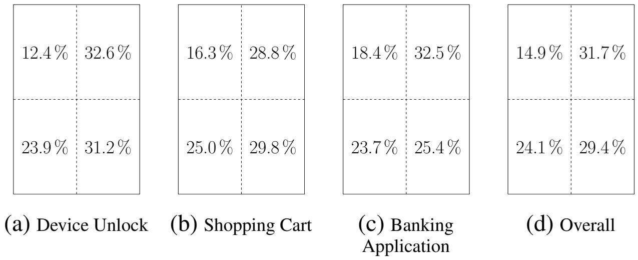
(a) Device Unlock (b) Shopping Cart (c) Banking Application (d) Overall

Figure 12: Frequency of start quadrants per scenario for the blocklist treatment.



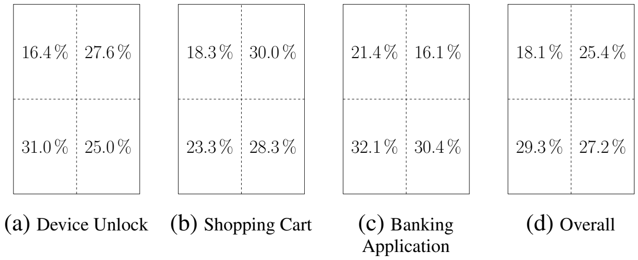
(a) Device Unlock (b) Shopping Cart (c) Banking Application (d) Overall

Figure 14: Frequency of start quadrants per scenario for the big treatment.



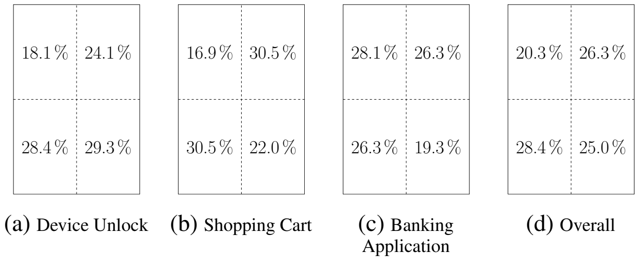
(a) Device Unlock (b) Shopping Cart (c) Banking Application (d) Overall

Figure 9: Frequency of end quadrants per scenario in the preliminary study.



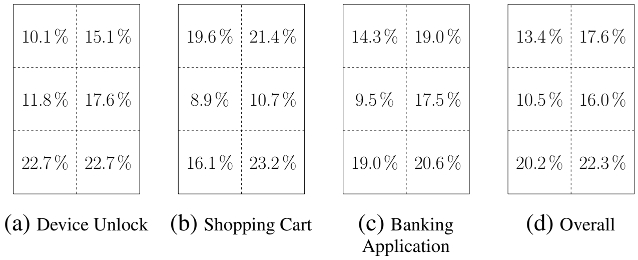
(a) Device Unlock (b) Shopping Cart (c) Banking Application (d) Overall

Figure 11: Frequency of end quadrants per scenario for the control treatment.



(a) Device Unlock (b) Shopping Cart (c) Banking Application (d) Overall

Figure 13: Frequency of end quadrants per scenario for the blocklist treatment.



(a) Device Unlock (b) Shopping Cart (c) Banking Application (d) Overall

Figure 15: Frequency of end quadrants per scenario for the big treatment.

Table 8: Comparison of the guessing metrics for a perfect-knowledge attacker between the scenarios.

	Dataset	Online Guessing (Success %)			Offline Guessing (bits)			
		λ_3	λ_{10}	λ_{30}	H_∞	$\widetilde{G}_{0.1}$	$\widetilde{G}_{0.2}$	$\widetilde{G}_{0.5}$
Control	Device Unlock*	17.9 %	37.5 %	73.2 %	3.81	3.81	4.10	4.93
	Banking App.	10.7 %	30.4 %	66.1 %	4.81	4.81	4.81	5.31
	Shopping Cart*	21.4 %	42.9 %	78.6 %	2.81	2.81	3.75	4.63
Blocklist	Device Unlock*	10.7 %	25.0 %	60.7 %	4.81	4.81	5.19	5.53
	Banking App.*	8.9 %	21.4 %	57.1 %	4.22	5.20	5.52	5.67
	Shopping Cart*	12.5 %	26.8 %	62.5 %	4.22	4.58	4.99	5.45
Large	Device Unlock*	17.9 %	41.1 %	76.8 %	3.81	3.99	4.20	4.79
	Banking App.*	12.5 %	32.1 %	67.9 %	4.22	4.58	4.68	5.21
	Shopping Cart	16.1 %	38.0 %	73.2 %	3.81	3.99	4.40	4.96
1st-Entry	Device Unlock*	16.1 %	33.9 %	69.6 %	3.81	3.99	4.40	5.12
	Banking App.*	12.5 %	28.6 %	64.3 %	4.22	4.58	4.81	5.38
	Shopping Cart.*	16.1 %	33.9 %	69.6 %	3.81	3.99	4.40	5.12

∗: For a fair comparison we downsampled all marked datasets to the size of the smallest datasets (56 Knock Codes).

Table 9: Qualitative codebook from post selection usability and security response.

Question	Code	Freq.	Description	Participant Sample
Security	RANDOM	78	Randomized use of quadrants and taps	"I tried to use random blocks to make it harder to guess."
	EASY TO REMEMBER	70	Prioritized memorability over security	"I was more concerned with it being easy to remember than security."
	LONG	56	Made codes longer as a means of security	"I tried to lengthen it to make it harder to crack."
	ALL QUADRANTS	52	Used all quadrants in the provided grid	"Using all the squares on all of the regions"
	UNEXPECTED	44	Avoided predictable patterns	"I tried to make it slightly more unpredictable than I normally would."
	NONE	42	Did not use any strategy for security	"Since this is not for my device I did not try to make it that secure. If it were my device I would write it down and it would be extensive."
	MULTIPLE QUADRANTS	40	Used a variety of quadrants, not necessarily all	"I tried to use multiple squares more than once to make it more secure."
	HARD TO GUESS	38	Chose a code that is difficult to guess	"Something I didn't think anyone could guess."
	DIFFERENT	37	Using a different code than the first one	"I needed it to be drastically different then [sic] the first code."
Memorability	PATTERN	104	Visualized a sequence or pattern	"Not overly random but three blocks of two patterns"
	SIMPLE	100	Used simple methods	"I used something that wasn't to [sic] complicated"
	EASY TO REMEMBER	77	Focused on overall memorability	"Something easy for me to remember but hard for someone else"
	DIRECTIONAL	76	Went in a specific sequence or order	"I used a specific direction as my way to remember like opening a box or lifting a lid."
	REPEATED	55	Tapped same quads multiple times	"I started at the top left quadrant and went clockwise."
	PERSONAL	52	Associated code with something personal	"I assigned numbers to the quadrants and input a date I'd remember."
	NONE	51	Had no strategy	"Didn't use one."
	VARIATION	40	Altered previous codes	"I used a combination that was similar to my other code but with a Twist."
	SHAPE	38	Followed a specific shape	"I patterned it off of a shape I would remember. In this case it was an underlined x."
	GAME	18	Used or made a game out of the sequence	"I tried to imagine a song pattern like Simon says."
Like	EASY	75	Found usability to be simple/straightforward	"Simple to input doesn't need much screen confirmation."
	HARD TO GUESS	42	Considered it a complex authentication	"I like how you can switch the codes up to many different patterns. It really makes it harder for people to guess what it is."
	DISCREET	40	Liked that it was/can be hidden and discrete	"You can be surreptitious and lock or unlock things without seeming like you are."
	QUICK	39	Found it to be efficient and quick	"It seems very convenient it can be quick and it gets old typing in my pin so much."
	FUN	32	Found it fun to use	"I like that they are unique and I like entering them it is enjoyable."
Dislike	HARD TO REMEMBER	124	Found it difficult to recall codes	"It's seems hard to remember the different patterns"
	INSECURE	90	Found it to be a less complex authentication	"Same thing as a pin without the numbers and with less combination possibilities."
	HARD TO TYPE	19	Found it difficult to input	"I could easily forget or tap the wrong location especially if there is no grid. Also it doesnt seem as fast as using a pattern to unlock like I currently do.."
	NONE	16	Had no issues	"Can't think of anything I overly dislike."
	NOT AN IMPROVEMENT	7	Considered it not better than other existing authentication methods	"There is absolutely no reason to use them for me or most people. They are hard to remember and not any different from a pin code."

Security: What strategy did you use to make your code **more memorable**?
Memorability: What strategy did you use to make your code **more secure**?
Like: What are some aspects you **like** about Knock Codes?
Dislike: What are some aspects you **do not like** about Knock Codes?

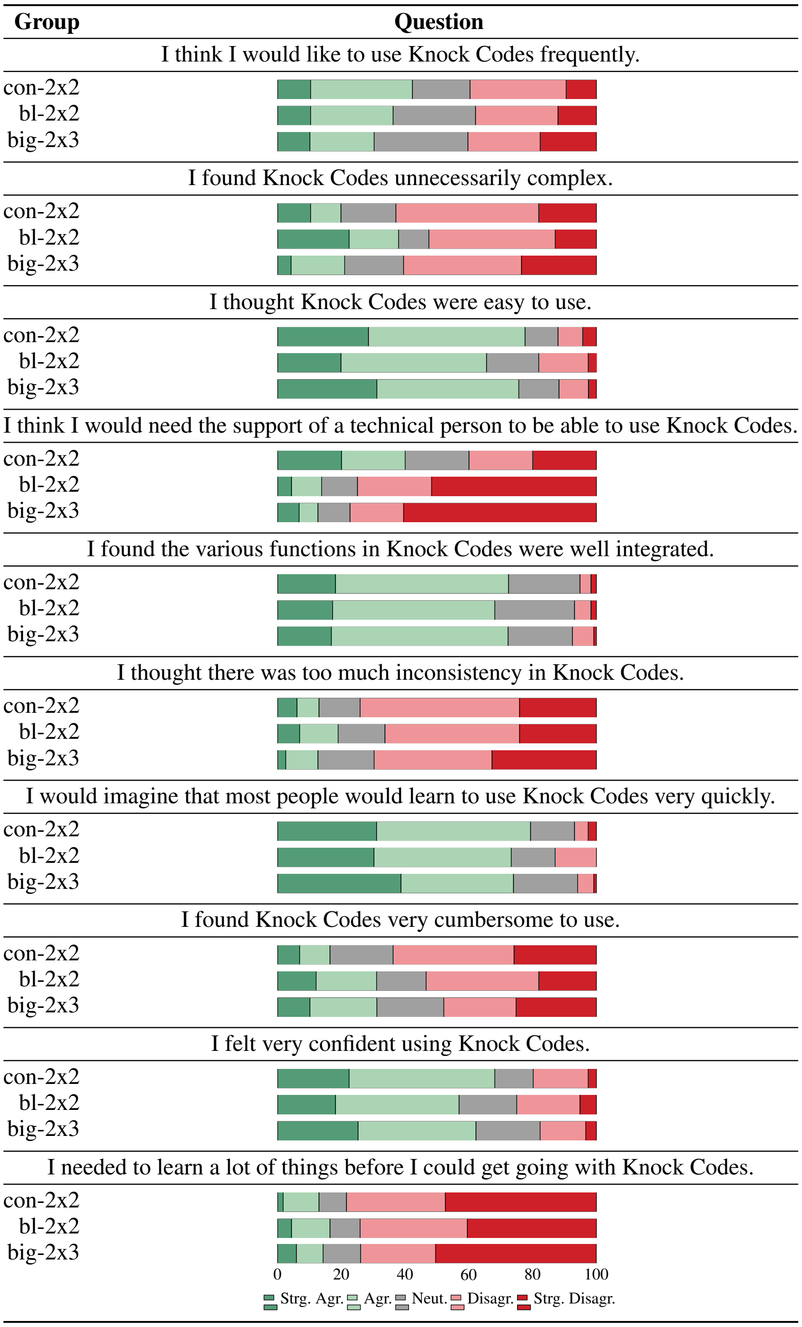


Figure 16: Responses to SUS questions: Averages are con-2x2 (69.8), bl-2x2 (65.3), and big-2x3 (68.1)